504-REN-BB-Specs

MISSION OPERATIONS AND DATA SYSTEMS DIRECTORATE

Advanced Composition Explorer Ground System Building Block Specifications: Data Services

October 1994

National Aeronautics and Space Administration

Goddard Space Flight Center
Greenbelt, Maryland

ACE Ground System Building Block Specifications: Data Services

Prepared for

GODDARD SPACE FLIGHT CENTER

By
COMPUTER SCIENCES CORPORATION

Under
Contract No. NAS5-31500

October 1994

Approved By:

(Name, Title) Date
(GSFC Organization or Company)

Branch Management Approval:

(Name, Branch Head or Tech Representative) Date
(GSFC Organization)

Division CCB Approval:

(Name, Chairman) Date
(GSFC Organization CCB)

Directorate CCB Approval:

(Name, Chairman) Date
(MO&DSD CCB)

Goddard Space Flight Center
Greenbelt, Maryland

Abstract

This document provides the building block specifications for the Advanced Composition Explorer project, as defined by the Renaissance Team Data Services Working Group.

Keywords:

Preface

This document is under the configuration management of the (organization) Configuration Control Board (CCB).

Configuration change requests to this document shall be submitted to the (organization) CCB, with supportive material justifying the proposed change. Changes to this document shall be made by document change notice or by complete revision.

Questions and proposed changes concerning this document shall be addressed to:

(Title) (Government Organization)
Code____
Goddard Space Flight Center
Greenbelt, Maryland 20771

Change Information Page

List of Effective Pages	
Page Number	**Issue**
All	Original

Document History			
Document Number	**Status/Issue**	**Publication Date**	**CCR Number**
	Original	October 1994	

DCN Control Sheet

DCN Number	Date/Time Group	Month/ Year	Section(s) Affected	Initials

Contents

Production Data Processing Subsystem (DS01)

DSN Monitor Block Processor (DS03)

Command Echo Processor (DS04)

Event Logger (DS05)

Telemetry Decommutation (DS07)

Data Server (DS08)

High-Speed Offline Subset (DS09)

File Server (DS10)

History Services (DS11)

Ingest Project Database (DS12a)

Extract Project Database (DS12b)

Database (DS13)

Real-Time Subset Generation (DS14)

Packet Server (DS15)

Data Distribution (DS19)

ACE TDM Processing (DS20)

CLCW Processing (DS21)

Abbreviations and Acronyms

Figures

Production Data Processing Subsystem (DS01)
Date of Specification: June 29, 1994

The development approach to the Production Data Processing Subsystem (PDPS) Building Block is to modify the existing Packet Processor II Data Capture Facility (Pacor II) PDPS.

1. Functional and Performance Overview

1.1 Functional Requirements Summary

1.1.1 The Advanced Composition Explorer (ACE) PDPS shall receive data (consisting of reassembled, annotated packet data) from Spacecraft Communications Services.

1.1.2 The PDPS shall perform level-zero processing (LZP) functions (time ordering based on the 32-bit spacecraft counter, overlap removal) and provide quality annotations.

1.1.3 The PDPS shall produce production data sets from playback data [virtual channel (VC) 2] and/or real-time data (VC1) over a designated time period (for example, from 12:00 a.m. to 11:59:59 p.m.).

1.1.4 The PDPS shall produce quick LZP of the playback (VC2) data from a single pass, as requested.

1.1.5 The PDPS shall produce quicklook data from each pass from the real-time stream (VC1).

1.2 Performance Requirements Summary

1.2.1 The PDPS shall ingest telemetry data at a rate equivalent to 76 kilobits per second (kbps).

1.2.2 The PDPS shall ingest replays of stored data at up to six times the original data rate (456 kbps).

1.2.3 The PDPS shall complete production data products within 12 hours of loss of signal of the pass containing the last bit to be included in the product.

[Reference: ACE Detailed Mission Requirements (DMR) (Draft), July 13, 1994, p 6200-1]

1.2.4 The PDPS shall complete quick LZP data products within 1.5 hours of the end of the pass or from the time of the request, whichever is later.

1.2.5 The PDPS shall complete quicklook data products within 1.5 (TBR) hours of the end of the pass.

1.2.6 The PDPS shall receive, process, and forward up to 82.5 megabytes of data per day.

2. Specifications

2.1 Interface Specifications (Figure DS01-1)

2.1.1 The PDPS shall interface with the

a. Packet Extraction Building Block of Spacecraft Communications Services
b. Synchronized system clock
c. Event Logger Building Block
d. Database Building Block
e. State Manager Function
f. Data Distribution Building Block

2.1.2 The PDPS shall use the services of the software backplane to interface with other building blocks.

2.1.1 Input

2.1.1.1 Packet Extraction Building Block of Spacecraft Communications Services: The PDPS shall receive packets with quality annotations from the Packet Extraction Building Block.

2.1.1.2 System Clock: The PDPS shall receive the current time from the synchronized system clock.

[Reference: Pacor II Detailed Design Specification (DDS), May 1993]

2.1.1.3 Database Building Block: The PDPS shall receive acquisition setup, product specifications, and subsystem setup information from the Database Building Block.

(Reference: Pacor II DDS, May 1993)

2.1.1.4 State Manager Function: The PDPS shall receive directives from the State Manager Function. Directives to be received include

a. Shutdown
b. Terminate capture
c. Start preprocessing
d. Task restart
e. Terminate preprocessing
f. Delete session
g. Delete product
h. Start product
i. Change product specification
j. Retransmit a product
k. Remove a product specification
l. Remove mission product specifications
m. Recover

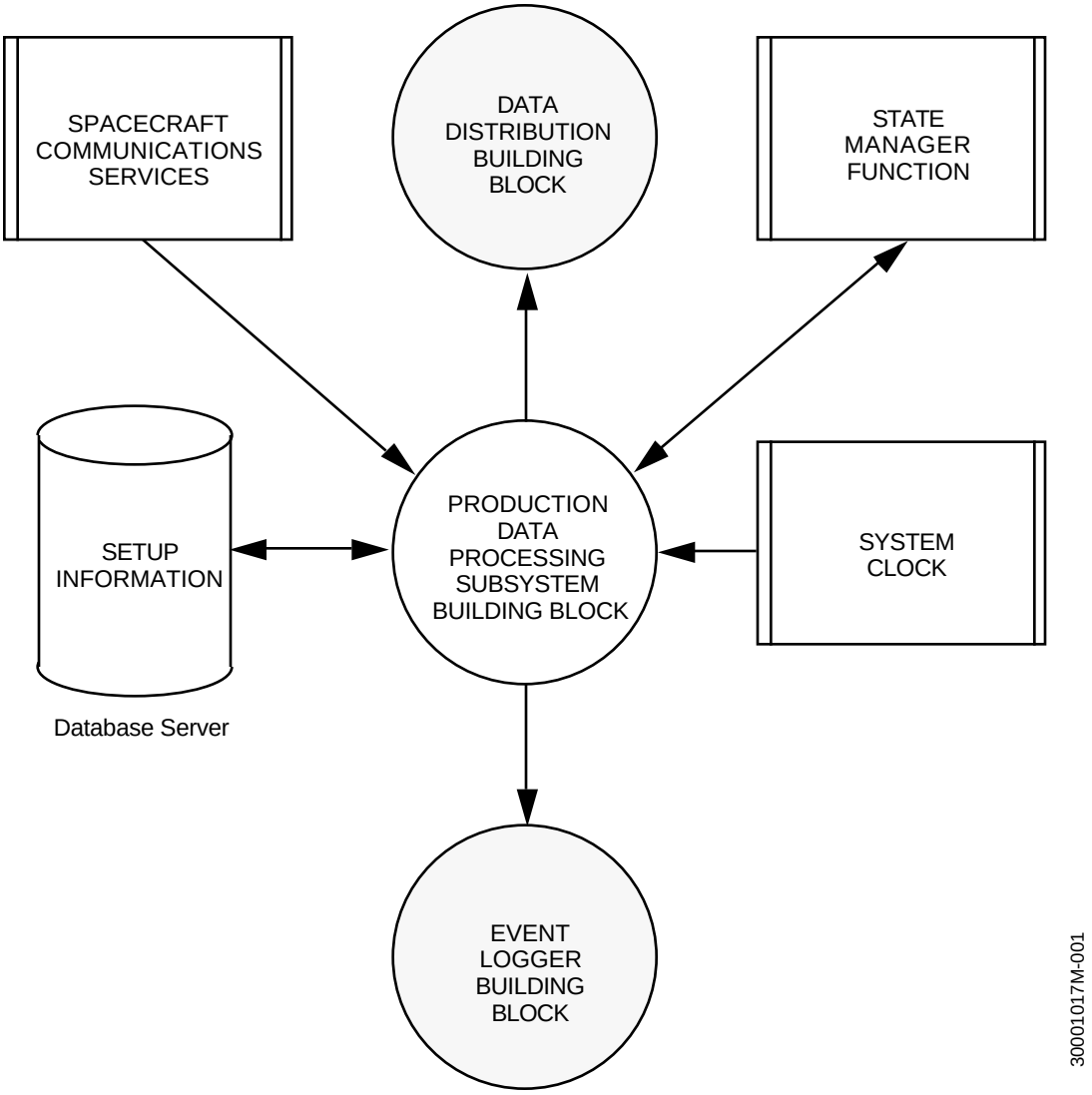

Figure DS01–1. Production Data Processing Subsystem Building Block Context

n. Include session in products
o. Check product threshold
p. New user product specification
q. Terminate product
r. Start log
s. Terminate log
t. Tape dump
u. Relocate session
v. Get session/product on disk

(Reference: Pacor II DDS, May 1993)

2.1.2 Output

2.1.2.1 Event Logger Building Block: The PDPS Building Block shall send a formatted event message consisting of an index to the event type, a criticality indicator, and the event message text. The receipt time should be stored in the first word of the event message text.

(Reference: Transportable Payload Operations Control Center (TPOCC) DDS for Release 10)

2.1.2.2 Database Building Block: The PDPS shall provide data capture status, product status, and other processing status to the Database Building Block.

(Reference: Pacor II DDS)

2.1.2.3 State Manager Function: The PDPS shall provide the following output to the State Manager Function:

a. Current status (on/off) for each of its functions
b. Acknowledgment of directives

2.1.2.4 Data Distribution Building Block: The PDPS shall provide LZP products with detached standard formatted data units (SFDUs) to the Data Distribution Building Block.

(Reference: Pacor II DDS)

2.1.3 Standards

The PDPS shall comply with the following standards:

a. Transmission Control Protocol/Internet Protocol (TCP/IP)
b. Portable Open System Interconnect Executive (POSIX.2)
c. Consultative Committee for Space Data Systems (CCSDS) Packet Telemetry
d. CCSDS SFDU
e. American National Standards Institute (ANSI) C
f. ANSI Structured Query Language (SQL)

2.1.4 Interface Performance Requirements

See performance requirements.

2.2 Functional Specifications

2.2.1 Required Use of Existing Components

a. Production Data Processing Software Control task
b. Capture acquisition
c. Reprocess acquisition session
d. Create grouped products

2.2.2 Functional Requirements Modifications

2.2.2.1 The PDPS shall execute concurrently with other applications on the same platform in a time-sharing environment.

2.2.2.2 The PDPS shall sort data using a non-CCSDS compatible mission elapsed time 32-bit counter that increments once every second. No subseconds field is provided or required.

[Reference: *ACE Command and Data Handling Component Specifications*, January 28, 1994 (Preliminary), p 166]

2.2.2.3 The PDPS shall be modified to handle a single mission that conforms to CCSDS recommendations. The code for supporting other telemetry formats [for example, Hubble Space Telescope (HST)] shall be removed.

2.2.2.4 The PDPS shall report events to the TPOCC Event Logger Building Block. Event formats shall be changed to be compatible with the Event Logger Building Block.

2.2.2.5 The PDPS shall interface with the TPOCC State Manager Function to receive directives and report current status.

2.2.3 Operational Requirements Modifications

2.2.3.1 Computer-Human Interface Requirements: Computer-human interface requirements will be implemented by user services.

2.2.3.2 Security Requirements: The PDPS shall comply with National Aeronautics and Space Administration (NASA) Automated Information System (AIS) Level 1 (TBR) security requirements to prevent unauthorized disclosure and/or modification of data and/or software.

3. Programmatic Requirements

3.1 Development Considerations

The PDPS Building Block shall conform to the TBS ACE development environment and to the standards listed in Section 2.1.3.

3.2 Portability

3.2.1 The PDPS Building Block shall be POSIX and ANSI C compliant.

3.2.2 The PDPS Building Block shall compile and execute on both the ACE real-time and workstation platforms.

3.2.3 The PDPS shall be transportable to platforms that adhere to the POSIX.2 and ANSI C standards.

3.3 Expandability

The existing PDPS can handle missions with up to 256 application identifiers (IDs) and 16 VC IDs.

3.4 Built-In Flexibility and Customization

The existing PDPS software was developed to support multiple missions that adhere to the CCSDS recommendations. It is sufficiently parameterized to support other missions that adhere to CCSDS recommendations. The PDPS software design allows for the use of mission-specific modules for handling non-CCSDS missions, such as the HST, or for handling special mission requirements.

DSN Monitor Block Processor (DS03)
Date of Specification: August 11, 1994

The development approach to the Deep Space Network (DSN) Monitor Block Processor Building Block is to modify the existing TPOCC DSN_MONITOR task.

Requirements: DS-020, DS-023, and DS-030

1. Functional and Performance Overview

The DSN Monitor Block Processor Building Block will receive DSN monitor blocks, extract parameters from the DSN blocks, and provide the parameters to the Data Server Building Block via the system variable table. The received data will be provided to the History Services Building Block for logging.

(Reference: TPOCC DDS for Release 10, Section 3.17)

1.1 Functional Requirements Summary

1.1.1 The DSN Monitor Block Processor Building Block shall provide the same functional capabilities that the TPOCC DSN_MONITOR task provides in processing NASA Communications (Nascom) blocks. This building block shall receive DSN monitor blocks through TCP/IP.

1.1.2 The DSN Monitor Block Processor Building Block shall

a. Combine information from the filter and the DSN monitor parameter tables to extract parameters from the DSN monitor blocks
b. Provide extracted parameters to the Data Server Building Block via the system variable data table
c. Provide received blocks to the History Services Building Block for logging before any processing is performed on the block

(Reference: TPOCC DDS for Release 10, Section 3.17.1)

1.2 Performance Requirements Summary

1.2.1 The DSN Monitor Block Processor Building Block shall be capable of receiving and processing DSN monitor blocks at a maximum rate of one block per second.

1.2.2 The DSN Monitor Block Processor Building Block shall distribute DSN parameters within 1 second of receiving the monitor block.

2. Specifications

2.1 Interface Specifications (Figure DS03-1)

2.1.1 The DSN Monitor Block Processor Building Block shall interface with the following building blocks, using TCP/IP socket connections:

a. Spacecraft Communications Services
b. Data Server Building Block
c. Event Logger Building Block (Reference: TPOCC DDS for Release 10, Section C 10.1.4)
d. History Services Building Block (Reference: TPOCC DDS for Release 10, Section C.9)

2.1.2 The DSN Monitor Block Processor Building Block shall interface with the system clock to retrieve the time.

2.1.3 The DSN Monitor Block Processor Building Block shall interface with the State Manager Function through the software backplane.

2.1.4 The DSN Monitor Block Processor Building Block shall interface with the database server via the software backplane.

2.1.1 Input

2.1.1.1 Spacecraft Communications Services: The DSN Monitor Block Processor Building Block shall receive the DSN monitor blocks from Spacecraft Communications Services through a TCP/IP socket connection.

2.1.1.2 State Manager Function: The DSN Monitor Block Processor Building Block shall receive directives and control commands (turn DSNMON on/off) from the State Manager Function through the software backplane.

2.1.1.3 System Clock: The DSN Monitor Block Processor Building Block shall retrieve the time from the synchronized system clock.

2.1.1.4 Database Server: The DSN Monitor Block Processor Building Block shall receive DSN filter and parameter information from the database server.

2.1.2 Output

2.1.2.1 State Manager Function: The DSN Monitor Block Processor Building Block shall send directive status messages and setup responses to the State Manager Function through the software backplane.

2.1.2.2 Data Server Building Block: The DSN Monitor Block Processor Building Block shall send extracted DSN parameters as determined by setup tables to the Data Server Building Block through the system variable table.

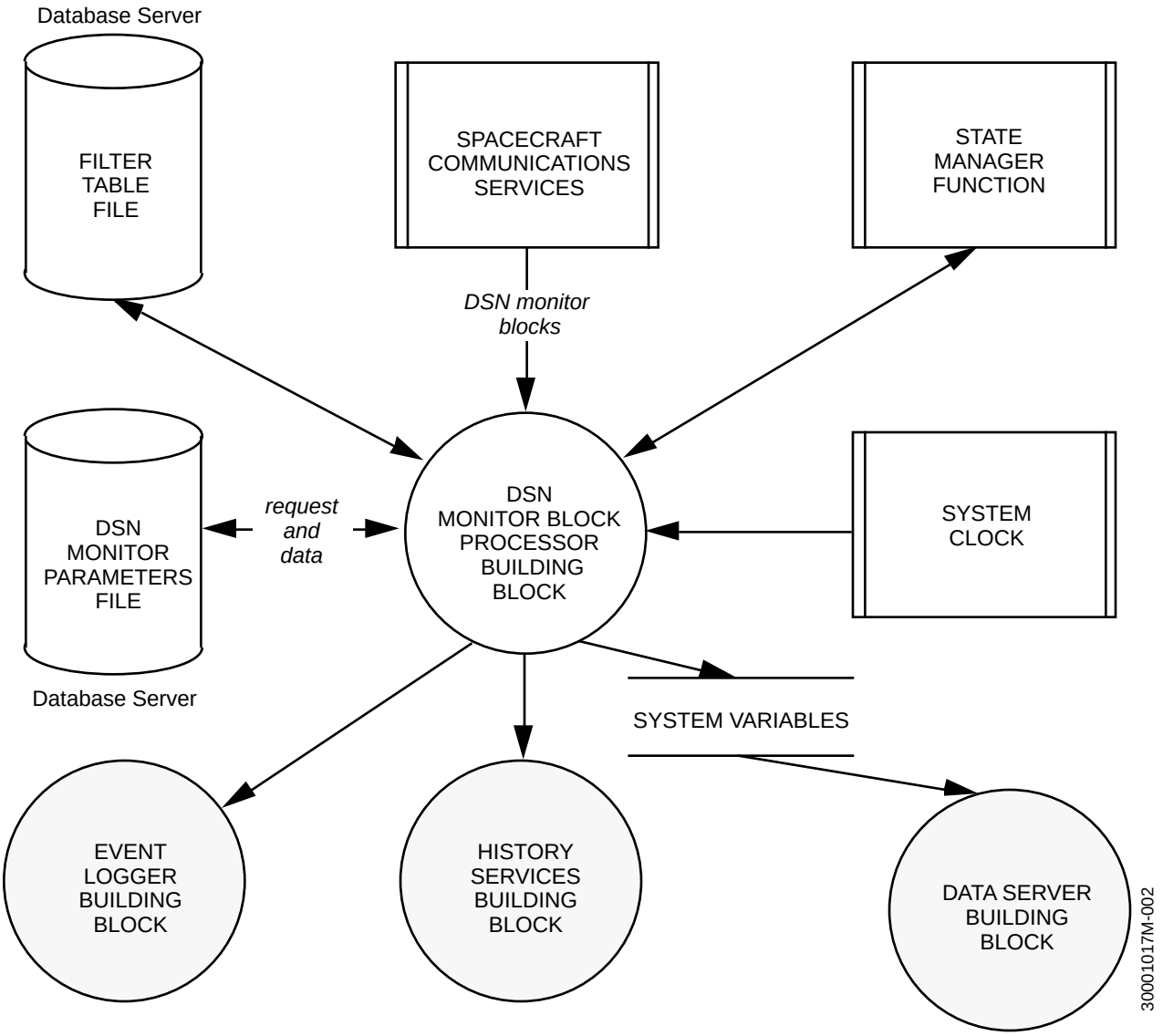

Figure DS03–1. DSN Monitor Processor Building Block Context

2.1.2.3 Event Logger Building Block: The DSN Monitor Block Processor Building Block shall send a formatted event message consisting of an index to the event type, a criticality indicator, and the event message text to the Event Logger Building Block. The receipt time should be stored in the first word of the event message text.

(Reference: TPOCC DDS for Release 10, Section C.10)

2.1.2.4 History Services Building Block: The DSN Monitor Block Processor Building Block shall forward received blocks, prior to any modifications, to the History Services Building Block for logging.

(Reference: TPOCC DDS for Release 10, Section C.9)

2.1.2.5 Database Server: The DSN Monitor Block Processor Building Block shall send requests for DSN filter and parameter information to the database server.

2.1.3 Standards

The DSN Monitor Block Processor Building Block shall comply with the following standards:

a. TCP/IP
b. POSIX.2/4
c. ANSI C
d. Network File System (NFS)
e. External Data Representation (XDR)
f. ANSI SQL

2.1.4 Interface Performance Requirements

2.1.4.1 Spacecraft Communications Services: The DSN Monitor Block Processor Building Block interface shall provide the capability to receive DSN blocks at a rate of one block per second.

2.1.4.2 State Manager Function: The DSN Monitor Block Processor Building Block shall interface with the State Manager Function through the software backplane.

2.1.4.3 Data Server Building Block: The DSN Monitor Block Processor Building Block interface shall provide the capability to distribute DSN parameters within 1 second of receiving the monitor block.

2.1.4.4 Event Logger Building Block: The DSN Monitor Block Processor Building Block interface to the Event Logger Building Block shall perform as specified in the TPOCC DDS for Release 10, Sections 3.10.2 and C.10.

2.1.4.5 System Clock: The system time will be available from the system clock on demand.

2.1.4.6 History Services Building Block: The DSN Monitor Block Processor Building Block interface to the History Services Building Block shall be capable of transferring blocks at a rate of one block per second.

2.2 Functional Specifications

2.2.1 Required Use of Existing Components

The DSN Monitor Block Processor Building Block shall be based on the TPOCC DSN_MONITOR task.

(Reference: TPOCC DDS for Release 10, Section 3.17.2)

2.2.2 Functional Requirements Modifications

The DSN Monitor Block Processor Building Block shall be modified to provide or delete the following functions:

a. Receive DSN monitor blocks from Spacecraft Communications Services through TCP/IP.
b. Extract parameters from the DSN monitor block.
c. Distribute (pass) specified parameters to the Data Server Building Block through the system variable table.
d. Send DSN monitor blocks to the History Services Building Block for logging before any processing is performed on the blocks.
e. Receive setup parameters from the Database Building Block.

The DSN Monitor Block Processor Building Block shall not use the TPOCC Nascom interface (TNIF).

2.2.3 Data Management Requirements Changes

Data management requirements changes are not applicable for this building block.

2.2.4 Performance Requirements Modifications

The DSN Monitor Block Processor Building Block shall be capable of receiving DSN monitor blocks at a rate of one block per second.

2.2.5 Operational Requirements Modifications

The DSN Monitor Block Processor Building Block does not require modifications to operational requirements.

3. Programmatic Requirements

3.1 Development Considerations

The DSN Monitor Block Processor Building Block shall conform to the TBS ACE development environment and to the standards listed in Section 2.1.3.

3.2 Portability

The DSN Monitor Block Processor Building Block shall be POSIX and ANSI C compliant, and shall compile and execute on both the ACE real-time and workstation platforms.

3.3 Expandability

The DSN Monitor Block Processor Building Block shall be capable of growth in terms of size of monitor blocks, number of parameters contained in those blocks, and rate at which the blocks are received.

3.4 Built-In Flexibility and Customization

The DSN Monitor Block Processor Building Block shall be configured based on database tables. The configuration parameters shall include the following:

a. Format of header structure
b. Format of trailer, as applicable
c. Size of monitor block
d. Size of individual parameters
e. Format of individual parameters
f. Parameters to be passed directly to clients

Command Echo Processor (DS04)
Date of Specification: August 10, 1994

The development approach to the Command Echo Processor Building Block is to develop a new design in accordance with current TPOCC architecture.

Requirements: DS-020, DS-023, and DS-031

1. Functional and Performance Overview

The Command Echo Processor Building Block will receive command echo messages from Spacecraft Communications Services through a TCP/IP socket connection. The source of the received message shall be validated, the message shall be logged before being forwarded to the State Manager Function, and a count of received messages shall be maintained.

1.1 Functional Requirements Summary

The Command Echo Processor Building Block shall

a. Receive command echo messages
b. Validate message header for source of message
c. Time tag received messages
d. Maintain count of received messages
d. Send command echo messages to the History Services Building Block for logging
e. Send command echo messages to the State Manager Function

1.2 Performance Requirements Summary

The Command Echo Processor Building Block shall receive command echo messages from Spacecraft Communications Services at a equivalent rate of 1 kbps, corresponding to the ACE command data rate.

2. Specifications

2.1 Interface Specifications (Figure DS04-1)

2.1.1 The Command Echo Processor Building Block shall interface with the following building blocks, using TCP/IP socket connections:

a. Spacecraft Communications Services
b. Event Logger Building Block (Reference: TPOCC DDS for Release 10, Section C10.1.4)
c. History Spacecraft Building Block (Reference: TPOCC DDS for Release 10, Section C.9)

2.1.2 The Command Echo Processor Building Block shall interface with the system clock to retrieve the time.

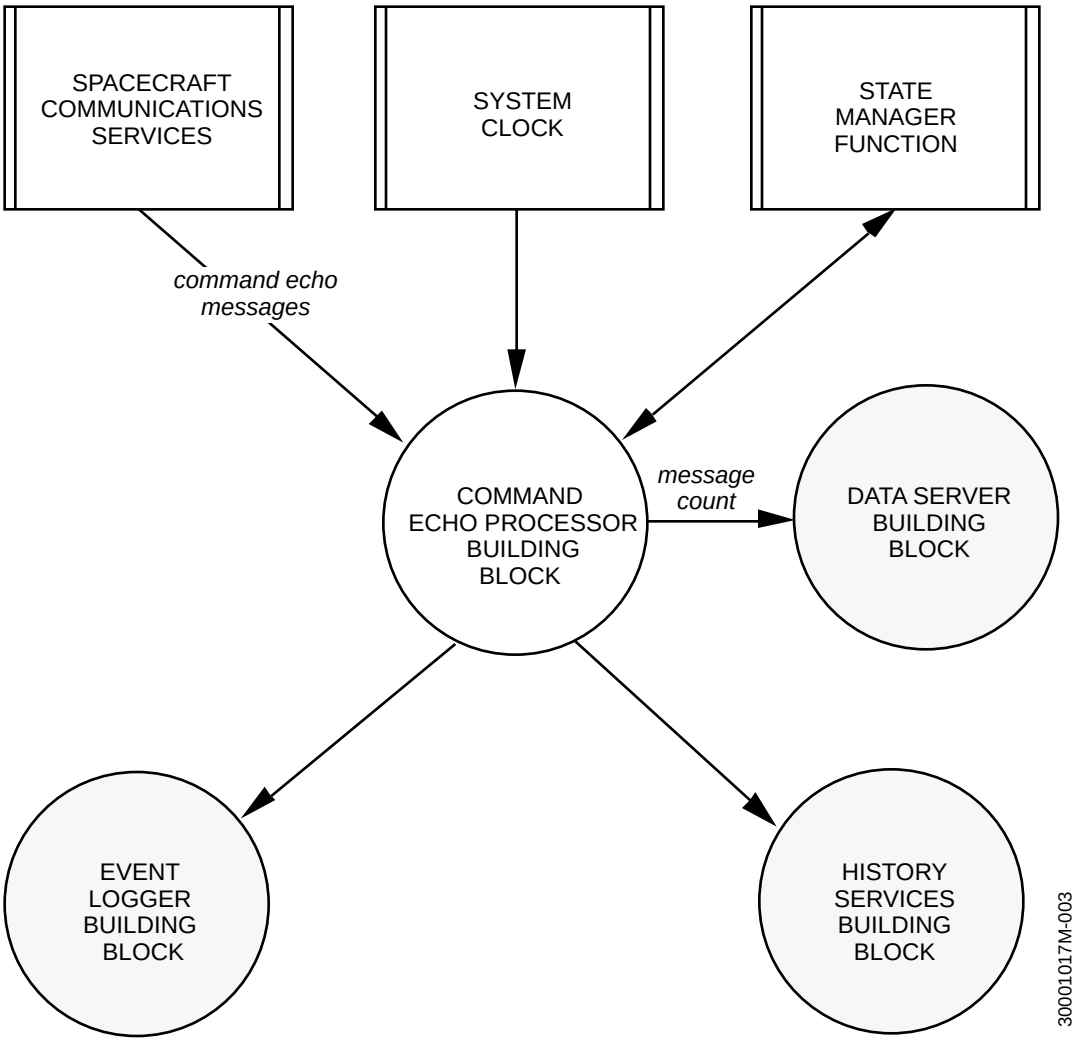

Figure DS04–1. Command Echo Processor Building Block Context

2.1.3. The Command Echo Processor Building Block shall interface with the State Manager Function through the software backplane.

2.1.4 The Command Echo Processor Building Block shall provide statistics to the Data Server Building Block via the system variable table.

2.1.1 Input

2.1.1.1 The Command Echo Processor Building Block shall receive command echo messages from Spacecraft Communications Services through a TCP/IP socket connection.

2.1.1.2 The Command Echo Processor Building Block shall receive setup directives (turn command echo processing on/off) from the State Manager Function through the software backplane.

2.1.1.3 The Command Echo Processor Building Block shall retrieve the time from the synchronized system clock.

2.1.2 Output

2.1.2.1 The Command Echo Processor Building Block shall send directive status messages to the State Manager Function through the software backplane.

2.1.2.2 The Command Echo Processor Building Block shall send formatted event messages consisting of an index to the event type, a criticality indicator, and the event message text to the Event Logger Building Block. The receipt time should be stored in the first word of the event message.

(Reference: TPOCC DDS for Release 10, Section C.10.1.4)

2.1.2.3 The Command Echo Processor Building Block shall send received command echo messages to the History Services Building Block for logging.

(Reference: TPOCC DDS for Release 10, Section C.9)

2.1.2.4 The Command Echo Processor Building Block shall provide the count of received command messages to the Data Server Building Block via the system variable table.

2.1.3 Standards

The Command Echo Processor Building Block shall comply with the following standards:

a. TCP/IP
b. POSIX.2/4
c. ANSI C
d. NFS
e. XDR

2.2 Functional Specifications

2.2.1 Functional Requirements

The Command Echo Processor Building Block shall

a. Interface with
    1. Spacecraft Communications Services
    2. State Manager Function through the software backplane
    3. Event Logger Building Block (Reference: TPOCC DDS Release 10, Section C.10.1.4)
    4. History Services Building Block (Reference: TPOCC DDS Release 10, Section C.9)
    5. System clock
    6. System variable table
b. Receive command echo messages from Spacecraft Communications Services
c. Time tag command echo messages
d. Validate message header for source of message
e. Generate event messages for each received command echo message and each invalid message
f. Accept directives from the State Manager Function through the software backplane
g. Retrieve the time from the system clock
h. Forward time-tagged, verified command echo messages to the History Services Building Block
i. Maintain count of received messages
j. Provide the count of received messages to the Data Server Building Block via the system variable table

2.2.3 Performance Requirements

The Command Echo Processor Building Block shall receive command echo messages from Spacecraft Communications Services at an equivalent rate of 1 kbps.

2.2.4 Operational Requirements Modifications

The Command Echo Processor Building Block does not require modifications to operational requirements.

3. Programmatic Requirements

3.1 Development Considerations

The Command Echo Processor Building Block shall conform to the ACE development environment and to the standards listed in Section 2.1.3.

3.2 Portability

The Command Echo Processor Building Block shall be POSIX and ANSI C compliant, and shall compile and execute on both the ACE real-time and workstation platforms.

3.3 Expandability

The Command Echo Processor Building Block shall allow expansion in the effective rate up to 4 kbps, corresponding to 100-percent growth in the current Small Explorer command rate.

3.4 Built-In Flexibility and Customization

The Command Echo Processor Building Block shall parameterize the format and size of expected command echo messages.

Event Logger (DS05)
Date of Specification: August 10,1994

The development approach to the Event Logger Building Block is to modify the existing TPOCC Event Logger, Event Server task, and New Event History task.

Requirements: DS-020, DS-023, and DS-032

1. Functional and Performance Overview

The Event Logger Building Block will receive event messages from various building blocks and then time-stamp and log messages to the event history file. It will provide an event server function that streams event messages to other building blocks based on specified filter parameters and store the information in the event database through the database management system (DBMS) interface. The Event Logger Building Block will stream filtered delogged event messages based on the requested event types, event numbers, criticality, and/or date and time. The Event Server task will provide real-time processing of events; the New Event History task will provide nonreal-time (NRT) processing of event information in the event database.

(Reference: TPOCC DDS for Release 10, Section 3.10)

1.1 Functional Requirements Summary

The Event Logger Building Block shall be capable of

a. Receiving formatted event messages from other building blocks through TCP/IP socket connections
b. Storing received event messages in an event database through the DBMS interface
c. Automatically inserting a time tag in the event history header if one is not present
d. Streaming filtered event messages to client building blocks
e. Receiving filter criteria from client building blocks
f. Receiving directives from the State Manager Function through the software backplane
g. Sending directive responses to the State Manager Function through the software backplane
h. Delogging from the event database based on filter criteria through the DBMS interface

(Reference: TPOCC DDS for Release 10, Section 3.10)

1.2 Performance Requirements Summary

1.2.1 The Event Logger Building Block shall receive, process, log, delog, and distribute events at a TBS rate.

1.2.2 The Event Logger Building Block shall write events to the event database every 15 seconds through the DBMS interface.

2. Specifications

2.1 Interface Specifications (Figures DS05-1 and DS05-2)

2.1.1 The Event Logger Building Block interface is a generic interface with other building blocks and shall follow the specifications of the TPOCC Event Logger interface described in the TPOCC DDS for Release 10, Sections 3.10.1 and C.10.

2.1.2 The Event Logger Building Block shall interface with the State Manager Function through the software backplane.

2.1.3 The Event Logger Building Block shall interface with the event database through the DBMS interface.

2.1.1 Input

a. Real-time events from other building blocks
b. Event type definitions from a system memory table
c. Setup directives from State Manager Function through the software backplane
d. Client requests from User Services (reports and displays)
e. Event type definitions from the event database through the DBMS interface
f. Historical events

(Reference: TPOCC DDS for Release 10, Section C.10)

2.1.2 Output

a. Filtered real-time events to User Services (client processes)
b. Updated event history records to DBMS interface
c. Setup responses to State Manager Function through the software backplane
d. Filtered historical event responses to User Services

(Reference: TPOCC DDS for Release 10, Section C.10)

2.1.3 Standards

The Event Logger Building Block shall comply with the following standards:

a. TCP/IP
b. POSIX.2/4
c. ANSI C
d. NFS
e. XDR
f. ANSI SQL

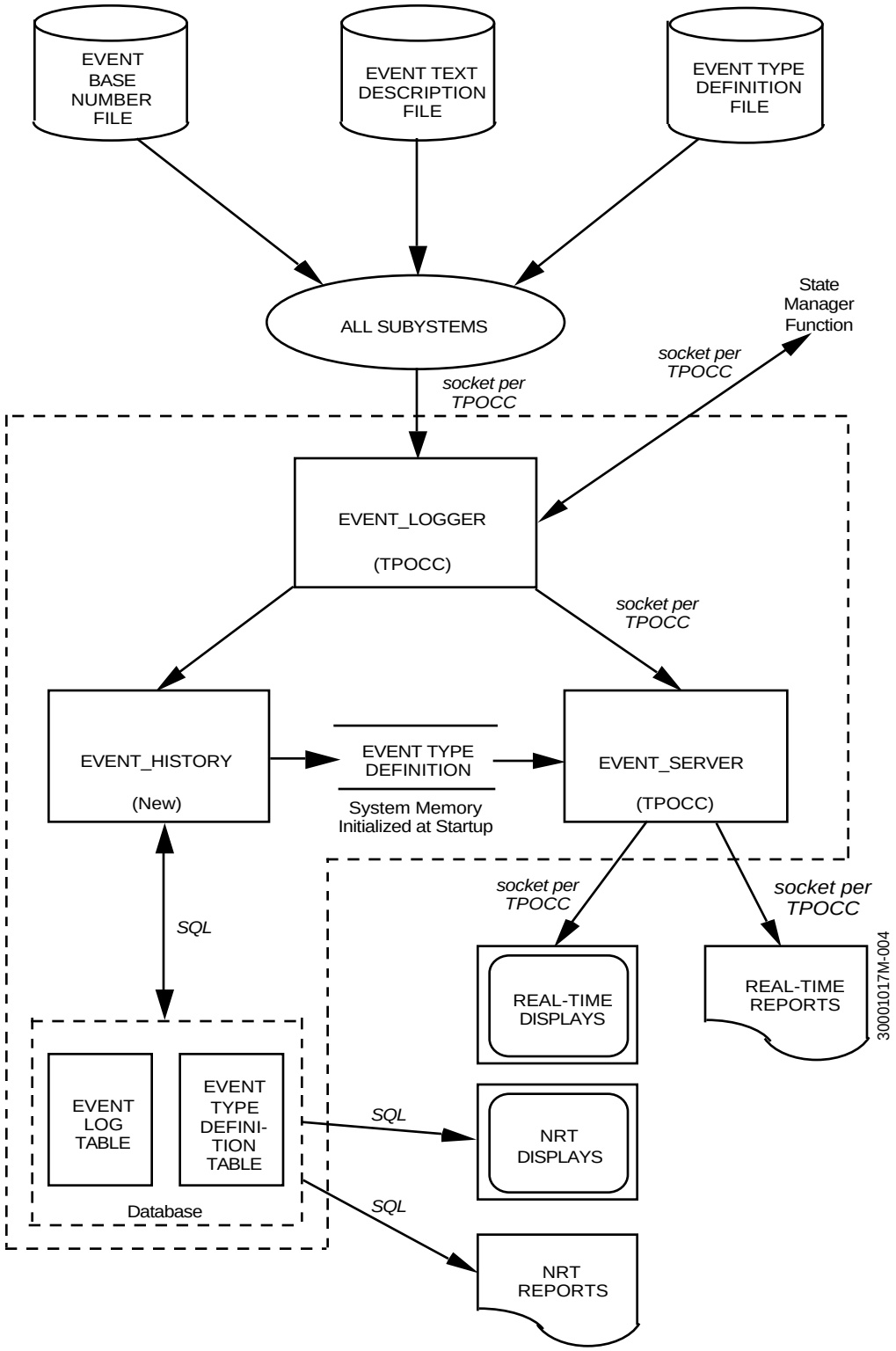

Figure DS05–1. Event Subsystem

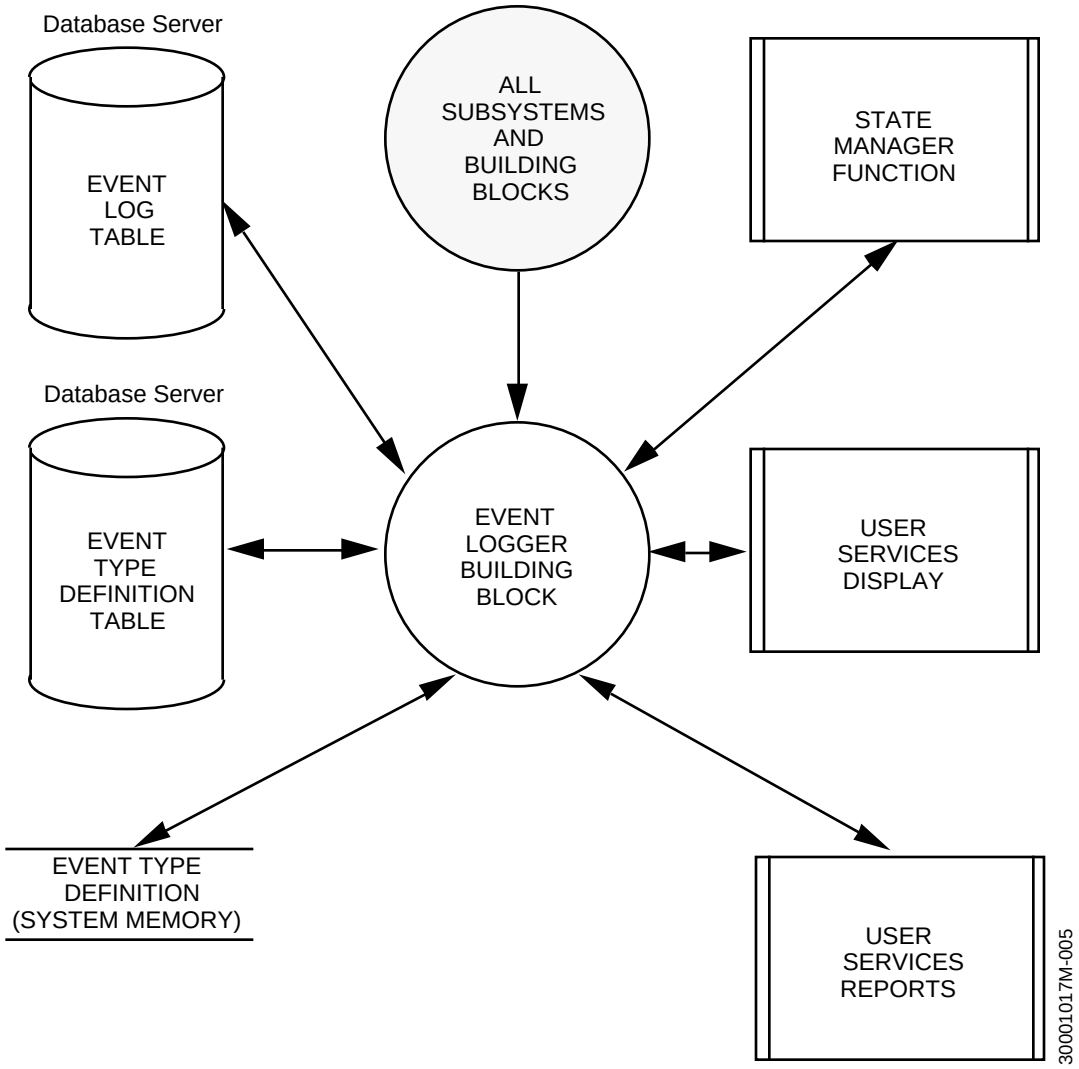

Figure DS05–2. Event Logger Building Block Context

2.1.4 Interface Performance Requirements

The Event Logger interfaces shall be capable of receiving, processing, logging and distribute events at a TBS rate.

2.2 Functional Specifications

2.2.1 Existing Building Block Reference

The following tasks of the TPOCC event subsystem shall be used:

a. EVENT_LOGGER
b. EVENT_SERVER

(Reference: TPOCC DDS for Release 10, Sections 3.10 and C.11)

NOTE: The New Event History task will provide NRT processing of event information in the event database through the DBMS interface.

2.2.2 Customization Provisions

Customization provisions are not applicable for this building block.

3. Programmatic Requirements

3.1 Development Considerations

The Event Logger Building Block shall conform to the ACE development environment and to the standards listed in Section 2.1.3.

3.2 Portability

The Event Logger Building Block shall be POSIX and ANSI C compliant, and shall compile and execute on both the ACE real-time and workstation platforms.

3.3 Expandability

Currently, the Event Logger Building Block has no expandability requirements.

3.4 Built-In Flexibility and Customization

The Event Logger Building Block shall be completely configured by database tables, setup directives, client requests, and/or compiler switches.

Telemetry Decommutation (DS07)
Date of Specification: August 10, 1994

The development approach to the Telemetry Decommutation Building Block is to modify the existing TPOCC TLM_DECOM task.

Requirements: DS-001, DS-013, DS-016, DS-020, and DS-023

1. Functional and Performance Overview

The Telemetry Decommutation Building Block provides the capability to decommutate telemetry data from a stream of minor frames of telemetry. It will provide the capability to decommutate nonsimultaneously both real-time and playback telemetry. The decommutation capability includes the ability to perform limit checking, annotate telemetry data with quality of engineering and attitude units, maintain counts of minor and major frames, generate event messages when limit violations occur, perform engineering unit conversions, and provide synchronous data service of decommutated telemetry points. The Telemetry Decommutation Building Block will be capable of handling data rates up to six times the playback rate of 76 kbps. The Telemetry Decommutation Building Block will use TCP/IP socket connections to interface with other building blocks. The TPOCC TLM_DECOM task will be used as the baseline for the Telemetry Decommutation Building Block.

(Reference: TPOCC DDS for Release 10, Section 3.5.2.1)

1.1 Functional Requirements Summary

1.1.1 The Telemetry Decommutation Building Block shall provide the same functional capabilities that the TPOCC TLM_DECOM task provides.

1.1.2 The Telemetry Decommutation Building Block shall

a. Combine information from the telemetry database and system variable database to extract telemetry points
b. Decommutate engineering, science housekeeping, and attitude data
c. Accept control directives
d. Accept telemetry data at up to six times the maximum input rate
e. Send data to the Data Server Building Block
f. Send event messages to the Event Server Building Block
g. Optionally perform limit checks
h. Validate receipt of minor frame data based on header information
i. Optionally perform engineering unit conversion when appropriate [Reference: *ACE Command and Data Handling Component Specifications* (Preliminary), January 28, 1994]
j. Annotate decommutated telemetry data with data quality
k. Maintain counts of major and minor frames in the system variable table

(Reference: TPOCC DDS for Release 10, Section 3.5.2.1)

1.2 Performance Requirements Summary

The Telemetry Decommutation Building Block shall be capable of handling the telemetry formats for engineering, science housekeeping, image dumps, and attitude data at data rates up to six times the real time specified for the ACE project (456 kbps).

(References: *ACE Command and Data Handling Component Specifications* (Preliminary), January 28, 1994, and ACE Telemetry Frame Format Descriptions)

2. Specifications

2.1 Interface Specifications (Figure DS07-1)

2.1.1 The Telemetry Decommutation Building Block shall interface with the following building blocks, using TCP/IP socket connections in a client/server environment:

a. Data Server Building Block (Reference: TPOCC DDS for Release 10, Sections C5.1.1 and C.11)
b. Packet Server Building Block (Reference: TPOCC DDS for Release 10, Section C7.4.1.3)
c. Event Logger Building Block (Reference: TPOCC DDS for Release 10, Section C 10.1.4)

(Reference: TPOCC DDS for Release 10, Sections C.5.1.2 and C.1.1.2)

2.1.2 The Telemetry Decommutation Building Block shall interface with the system clock to retrieve the time.

2.1.3 The Telemetry Decommutation Building Block shall interface with the State Manager Function through the software backplane.

2.1.4 The Telemetry Decommutation Building Block shall interface with the database server.

2.1.1 Input

2.1.1.1 Data Server Building Block: The Telemetry Decommutation Building Block shall receive data service control commands (directives) from the Data Server Building Block:

a. Request for telemetry mnemonic
b. Stop sending telemetry mnemonic
c. Mission-specific command

(Reference: TPOCC DDS for Release 10, Section 3.5.2.1)

2.1.1.2 Packet Server Building Block: The Telemetry Decommutation Building Block shall receive server messages from the Packet Server Building Block:

a. Telemetry packets in an ACE-specified CCSDS format (Reference: *ACE Command and Data Handling Component Specifications* (Preliminary), January 28, 1994)
b. Other server messages

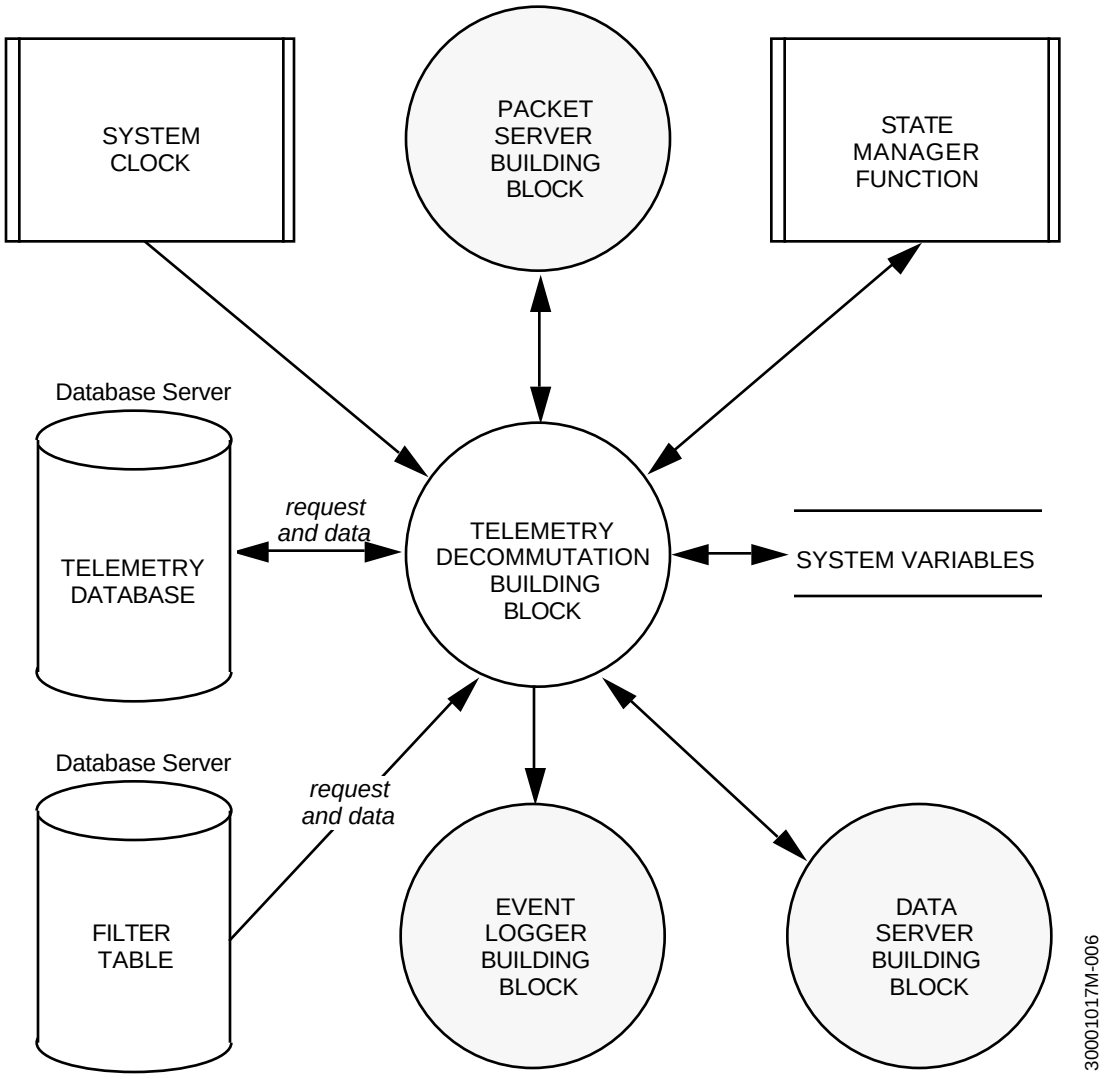

Figure DS07–1. Telemetry Decommutation Building Block Context

(Reference: TPOCC DDS for Release 10, Section C.7.4.1.3)

2.1.1.3 System Clock: The Telemetry Decommutation Building Block shall retrieve the time from the synchronized system clock.

2.1.1.4 State Manager Function: The Telemetry Decommutation Building Block shall receive the following directives and control commands from the State Manager Function through the software backplane:

a. Turn decommutation on/off
b. Turn engineering unit conversion on/off
c. Turn high and low limit checking on/off
d. Change high and low limit definition
e. Turn delta limit checking on/off
f. Turn decommutation on/off
g. Set up for pass initialization
h. Change coefficients for polynomial conversion
i. Change coordinates for straight line conversion

2.1.1.5 Database Server: The Telemetry Decommutation Building Block shall receive information from the telemetry database and system variable database used in extracting telemetry points from the database server through the software backplane.

2.1.2 Output

2.1.2.1 Data Server Building Block: The Telemetry Decommutation Building Block shall send selected filtered decommutation data (engineering, instrument health and safety, image dumps, and attitude) and client responses to the Data Server Building Block.

(Reference: TPOCC DDS for Release 10, Section C.11.1)

2.1.2.2 Packet Server Building Block: The Telemetry Decommutation Building Block shall send client messages to the Packet Server Building Block:

a. Packet service type
b. Filter table
c. Service activation
d. Service deactivation
e. Disconnection

(Reference: TPOCC DDS for Release 10, Section C.7.4.1.3)

2.1.2.3 Event Logger Building Block: The Telemetry Decommutation Building Block shall send formatted event messages consisting of an index to the event type, a criticality indicator, and the event message text to the Event Logger Building Block. The receipt time should be stored in the first word of the event message text.

(Reference: TPOCC DDS for Release 10, Section 3.10.2.1)

2.1.2.4 State Manager Function: The Telemetry Decommutation Building Block shall send directive status messages to the State Manager Function through the software backplane.

2.1.2.5 Database Server: The Telemetry Decommutation Building Block shall send a request for information to the database server through the software backplane.

2.1.3 Standards

The Telemetry Decommutation Building Block shall comply with the following standards:

a. TCP/IP
b. POSIX.2/4
c. ANSI C
d. NFS
e. XDR
f. Institute of Electrical and Electronics Engineers (IEEE) Floating Point
g. ANSI SQL

2.1.4 Interface Performance Requirements

2.1.4.1 The Data Server Building Block and Packet Server Building Block interfaces shall provide the capability to receive and send data in the format and data rates specified for the ACE project.

(Reference: *ACE Command and Data Handling Component Specifications* (Preliminary), January 28, 1994)

2.1.4.2 The Telemetry Decommutation Building Block interface to the Event Logger Building Block shall perform as specified in the TPOCC DDS for Release 10, Sections 3.10.2 and C.10.

2.1.4.3 The system time will be available from the system clock on demand.

2.1.4.4 The Telemetry Decommutation Building Block shall interface with the State Manager Function through the software backplane.

2.2 Functional Specifications

2.2.1 Required Use of Existing Components

The Telemetry Decommutation Building Block shall be based on the TPOCC TLM_DECOMMUTATION task.

(Reference: TPOCC DDS for Release 10, Section 3.5.2.1)

2.2.2 Functional Requirements Modifications

The Telemetry Decommutation Building Block shall be modified to provide the following functions:

a. Ingest ACE real-time and playback data formats

b. Recognize ACE minor and major frames, and maintain counts of frames
c. Perform decommutation based on ACE major frames
d. Decommutate engineering, instrument health and safety, attitude, and image dump data
e. Annotate telemetry data with quality of engineering and attitude units

[Reference: *ACE Command and Data Handling Component Specifications* (Preliminary), January 28, 1994]

2.2.3 Data Management Requirements Changes

Data management requirements changes are not applicable for this building block.

2.2.4 Performance Requirements Modifications

The Telemetry Decommutation Building Block shall be capable of processing ACE telemetry data at six times the playback rate of 76 kbps (456 kbps).

2.2.5 Operational Requirements Modifications

Operational requirements will be provided by User Services.

3. Programmatic Requirements

3.1 Development Considerations

The Telemetry Decommutation Building Block shall conform to the ACE development environment and to the standards listed in Section 2.1.3.

3.2 Portability

The Telemetry Decommutation Building Block shall be POSIX and ANSI C compliant, and shall compile and execute on both the ACE real-time and workstation platforms.

3.3 Expandability

This section is to be supplied.

3.4 Built-In Flexibility and Customization

The following types of data shall be parameterized:

a. Size of header
b. Size of minor and major frames
c. Size of telemetry parameter definition list
d. Size of linear calibration definition list
e. Size of polynomial calibration definition list
f. Size of limits definition list
g. Size of delta limit definition list

h. Size of switch definition list
i. Size of telemetry parameter location list
j. Size of telemetry parameter location definition
k. Size of context switch definition
l. Size of context-dependent decommutation list
m. Size of decommutation information for bits in a minor frame word
n. Size of parameter build table
o. Size of records in telemetry database dump file
    1. Project database (PDB) administration
    2. Telemetry specification record
    3. Limits definition record
    4. Delta limit definition table
    5. Polynomial conversion definition record
    6. Straight-line conversion definition record
    7. Straight-line conversion switch definition table record
    8. Location definition record
    9 Context-dependent definition record
p. Size of filter table file records
    1. Filter bytes record
    2. Filter values record

Data Server (DS08)
Date of Specification: August 9, 1994

The development approach to the Data Server Building Block is to reuse the existing TPOCC DATA_SERVER task.

Requirements: DS-020 and DS-023

1. Functional and Performance Overview

The Data Server Building Block will route data from a host building block to other building blocks. Client building blocks send requests to the data server specifying their data needs. The data server requests the specified data from the data source and routes the data to the requesting building block.

(Reference: TPOCC DDS for Release 10, Section 3.11.2.1)

1.1 Functional Requirements Summary

1.1.1 The Data Server Building Block shall route data from the Telemetry Decommutation Building Block to other subsystems and building blocks, including the Real-Time Subset Building Block, User Services, and Applications Services.

1.1.2 The Data Server Building Block shall accept asynchronous requests that include one-shot updates or periodic updates.

1.1.3 The Data Server Building Block shall accept synchronous requests transferring all of the requested data types as they are received.

1.1.4 The Data Server Building Block shall be capable of distributing parameters contained in the system variable table.

(Reference: TPOCC Detailed Design Specifications for Release 10, Section 3.11.2.1)

1.2 Performance Requirements Summary

This section is to be supplied.

2. Specifications

2.1 Interface Specifications (Figure DS08-1)

The Data Services interface is a generic interface used to start and stop data updates from the data server by a client building block and from telemetry decommutation by the data server. The data server interface shall follow the specifications of the TPOCC Data Services interface described in the TPOCC DDS for Release 10, Sections 3.11.1.1 and C.11.1.

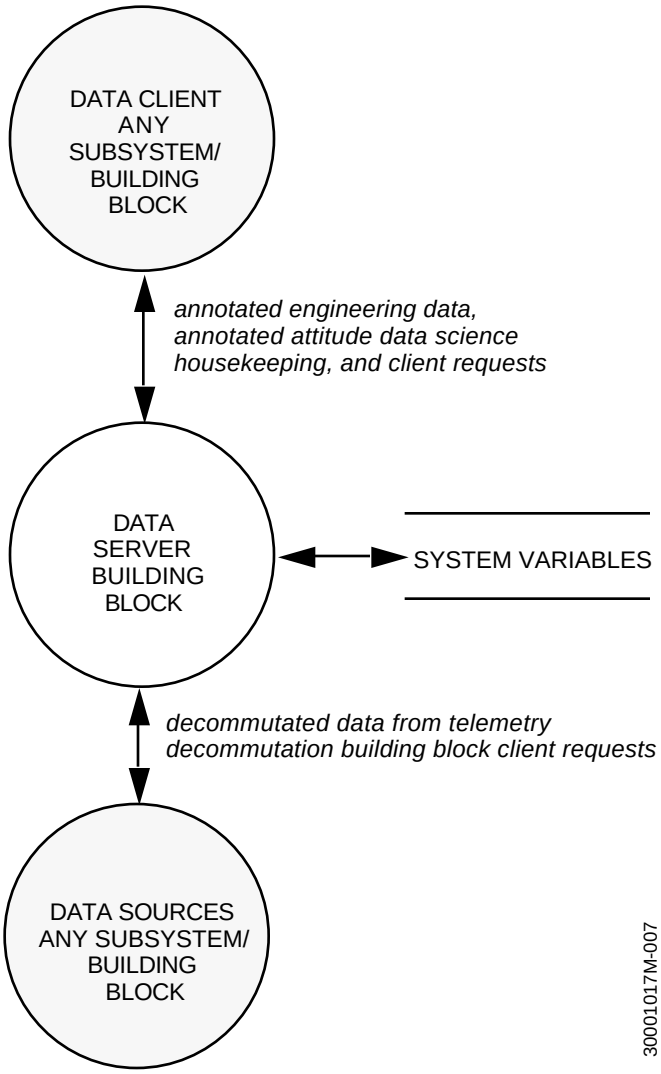

Figure DS08–1. Data Server Building Block Context

2.1.1 Input

a. Decommutated data from Telemetry Decommutation Building Block
b. System variables
c. Client requests (add, delete, or echo-request directives)

(Reference: TPOCC DDS for Release 10, Sections 3.11.1.1 and C.11.1)

2.1.2 Output

a. Requested data items (one-shot, periodic, and continuous)
b. Updates to system variables
c. Client responses to requesting building block (data, echo-answer, or end-of-history messages)

(Reference: TPOCC DDS for Release 10, Sections 3.11.1.1 and C.11.1)

2.1.3 Standards

The Data Server Building Block shall comply with the following standards:

a. TCP/IP
b. POSIX.2/4
c. ANSI C
d. XDR

2.1.4 Interface Performance Requirements

The data server interface shall be capable of receiving, processing, and distributing data at a TBS rate.

2.2 Functional Specifications

2.2.1 Existing Building Block Reference

The DATA_SERVER task of the TPOCC event subsystem shall be used.

(Reference: TPOCC DDS for Release 10, Sections 3.11 and C.11)

2.2.2 Customization Provisions

Customization provisions are not applicable to this building block.

3. Programmatic Requirements

3.1 Development Considerations

The Data Server Building Block shall conform to the ACE development environment and to the standards listed in Section 2.1.3.

3.2 Portability

The Data Server Building Block shall be POSIX and ANSI C compliant, and shall compile and execute on both the ACE real-time and workstation platforms.

3.3 Expandability

The Data Server Building Block shall interface with any building block, either as a server or a client, conforming to the interface specified in Section 2.2.1.

3.4 Built-In Flexibility and Customization

The data server shall be completely configured based on the system variable database and client directives.

High-Speed Offline Subset (DS09)
Date of Specification: August 10, 1994

Modification of the High-Speed Offline Subset Building Block is currently under development by Code 511.

Requirements: DS-020, DS-023, and DS-034

1. Functional and Performance Overview

The High-Speed Offline Subset Building Block will generate subset files directly from logged packets and/or LZP data sets from the ACE Production Processor System. This building block will allow the subsetting of one acquisition session within 2 hours of receipt of quick LZP products.

1.1 Functional Requirements Summary

The High-Speed Offline Subset Building Block shall

a. Interface with the File Server Building Block, State Manager Function through the software backplane, Event Logger Building Block (Reference: TPOCC DDS Release 10, Section C.10.1.4), system clock, and database server
b. Receive grouped product output data from the File Server Building Block
c. Produce binary subset data and subset summaries
d. Send formatted event messages consisting of an index to the event type, a criticality indicator, and the event message text (the receipt time should be stored in the first word of the event message text)
e. Receive directives from the State Manager Function through the software backplane
f. Send directive responses to the State Manager Function through the software backplane
g. Retrieve the time from the system clock on demand
h. Retrieve subset definitions from the database server

1.2 Performance Requirements Summary

The High-Speed Offline Subset Building Block shall

a. Receive grouped product output data from the File Server Building Block up to a TBS maximum file size
b. Provide subsets based on up to 2000 parameters per subset
c. Allow the subsetting of one acquisition session within 2 hours of receipt of quick LZP products

2. Specifications

2.1 Interface Specifications (Figure DS09-1)

2.1.1 The High-Speed Offline Subset Building Block shall interface with the File Server and Event Logger Building Blocks, using TCP/IP socket connections in a client/server environment.

(Reference: TPOCC DDS Release 10, Section C.10.1.4)

2.1.2 The High-Speed Offline Subset Building Block shall interface with the system clock.

2.1.3 The High-Speed Offline Subset Building Block shall interface with the State Manager Function through the software backplane.

2.1.4 The High-Speed Offline Subset Building Block shall interface with the database server through the software backplane.

2.1.1 Input

2.1.1.1 File Server Building Block: The High-Speed Offline Subset Building Block shall receive TBS from the grouped product output from the File Server Building Block.

2.1.1.2 State Manager Function: The High-Speed Offline Subset Building Block shall receive setup directives from the State Manager Function through the software backplane.

2.1.1.3 System Clock: The High-Speed Offline Subset Building Block shall retrieve the time from a synchronized system clock.

2.1.1.4 Database Server: The High-Speed Offline Subset Building Block shall retrieve subset definitions from the database server.

2.1.2 Output

2.1.2.1 State Manager Function: The High-Speed Offline Subset Building Block shall send directive status messages to the State Manager Function through the software backplane.

2.1.2.2 Event Logger Building Block: The High-Speed Offline Subset Building Block shall send formatted event messages consisting of an index to the event type, a criticality indicator, and the event message text to the Event Logger Building Block. The receipt time should be stored in the first word of the event message text.

(Reference: TPOCC DDS for Release 10, Section C.10.1.4)

2.1.2.3 Database Server: The High-Speed Offline Subset Building Block shall send binary subsets and subset summaries to the database server.

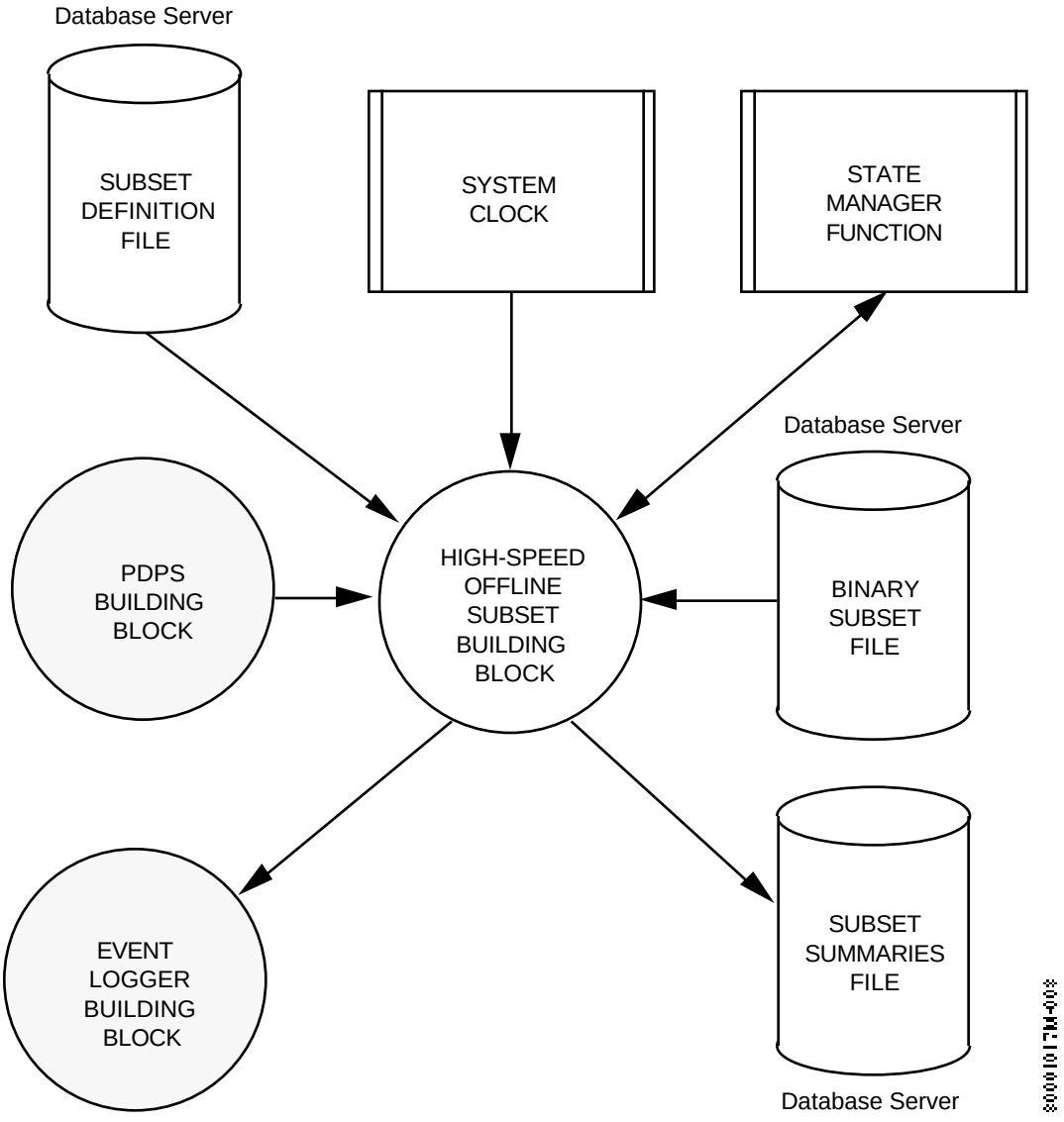

Figure DS09–1. High-Speed Offline Subset Building Block Context

2.1.3 Standards

The High-Speed Offline Subset Building Block shall comply with the following standards:

a. TCP/IP
b. POSIX.2
c. ANSI C
d. XDR
e. IEEE Floating Point
f. NFS
g. ANSI SQL

2.2 Functional Specifications

2.2.1 Required Use of Existing Components

The High-Speed Offline Subset Building Block shall be based on the equivalent TPOCC element currently in development.

2.2.2 Functional Requirements Modifications

In addition to the current requirements and capabilities, the High-Speed Offline Subset Building Block shall accept and process LZP.

2.2.3 Data Management Requirements Changes

Data management requirements changes are not applicable to this building block.

2.2.4 Performance Requirements Modifications

This building block has no identified performance requirements changes.

2.2.5 Operational Requirements Modifications

This building block has no identified operational requirements changes.

3. Programmatic Requirements

3.1 Development Considerations

The High-Speed Offline Subset Building Block shall conform to the ACE development environment as specified in the TBS and to the standards listed in Section 2.1.3.

3.2 Portability

The High-Speed Offline Subset Building Block shall be POSIX and ANSI C compliant.

3.3 Expandability

The High-Speed Offline Subset Building Block shall be expandable to allow subsets of information to TBS parameters.

3.4 Built-In Flexibility and Customization

The following types of data shall be parameterized:

a. Number of parameters per subset
b. Unique parameters to be included in each subset (by mnemonic)
c. Type of parameter (raw, converted, or both)
d. Telemetry parameter definitions
e. Calibration definitions
f. Limits definitions

File Server (DS10)
Date of Specification: August 16, 1994

The development approach for the File Server Building Block is commercial off-the-shelf (COTS) products and reuse.

1. Functional and Performance Overview

The File Server Building Block provides the ACE ground system with the capabilities for storage and retrieval, backup and recovery, and access control of ACE data files. The file server incorporates NFS standards and capabilities to allow remote mounting of disks and for standardizing data formats to allow transferring files among different hardware platforms.

The File Server Building Block stores the data files online for the required amount of time, backs them up on offline media for recovery in case of failure of the online media, provides access to them by authorized applications and users, and moves the files to archive media when the online time requirement has elapsed.

Security and access control capabilities provided by the file server are those provided by the host operating system and the ACE software backplane.

This building block combines COTS products and other reusable components from TBS.

1.1 Functional Requirements Summary

The File Server Building Block shall

a. Provide the remote access and data standardization capabilities of NFS
b. Provide capabilities to back up all files in online storage to backup media
c. Provide capabilities to restore files to online storage from backup media
d. Monitor the expiration date on all data files in online storage and delete any expired files from online media
e. Limit access to data files to authorized users and processes
f. Read required data file storage durations and user access privileges from the ACE database
g. Provide File Transfer Protocol (FTP) access to online data files
h. Answer requests to check the existence of data files

1.2 Performance Requirements Summary

1.2.1 The File Server Building Block shall be capable of handling the following files in the amounts indicated:

File Type	Maximum No. of Files
TBS	TBS

1.2.2 The File Server Building Block shall provide TBS bytes of online storage.

1.2.3 The File Server Building Block shall be capable of restoring an expired file from offline storage with 24 hours.

2. Specifications

2.1 Interface Specifications (Figure DS10-1)

2.1.1 The File Server Building Block shall interface with other ACE ground system building blocks through the ACE software backplane.

2.1.2 The File Server Building Block shall interface with the following building blocks:

a. Database
b. Event Logger
c. Any ACE building block

2.1.1 Input

2.1.1.1 Other ACE Building Blocks: The File Server Building Block shall accept the following from the other ACE building blocks through the ACE software backplane:

a. Requests to mount remote disk drives
b. Requests to create new files
c. Requests to read, write, or edit existing files
d. Files to be stored online
e. Requests to check existence of data files

2.1.1.2 Database Building Block: The File Server Building Block shall retrieve the following from the Database Building Block through the ACE software backplane:

a. Online storage durations for each data file type
b. Access privileges to data files

2.1.2 Output

2.1.1.1 Other ACE Building Blocks: The File Server Building Block shall provide to the other ACE ground system building blocks through the ACE software backplane:

a. Access to data files in online storage
b. FTP access to data files in online storage

2.1.1.2 Event Logging Building Block: The File Server Building Block shall output to the Event Logging Building Block through the ACE software backplane event messages identifying data sets that have expired, been moved to offline media, or been restored from offline media.

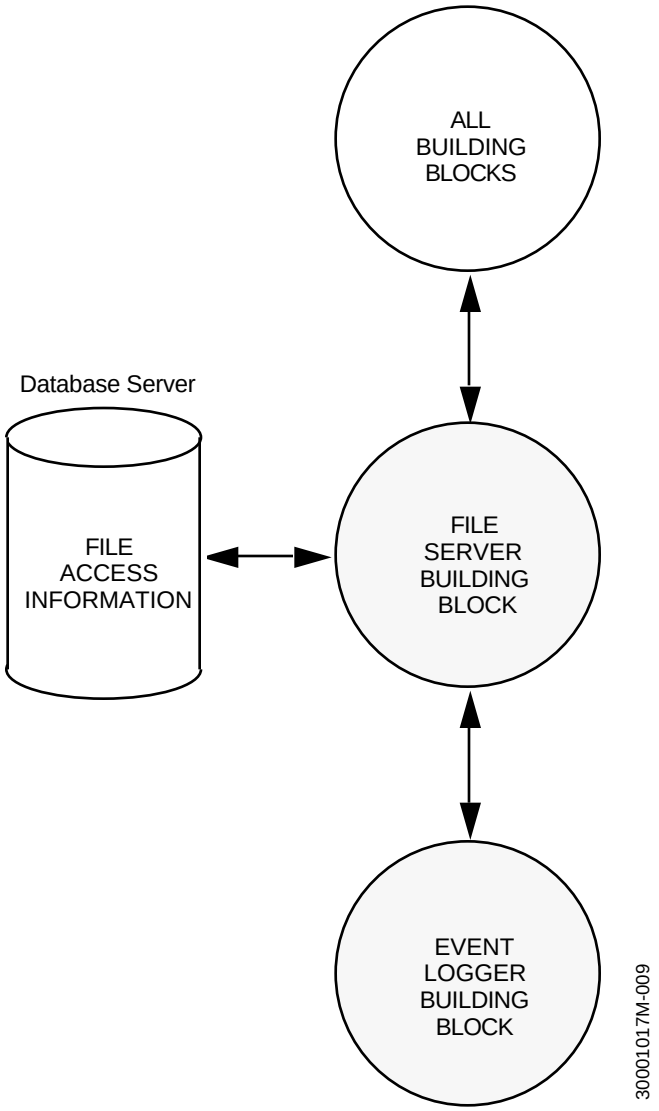

Figure DS10–1. File Server Building Block Context

2.1.3 Standards

The File Server Building Block shall comply with the following standards:

a. TCP/IP
b. POSIX
c. ANSI C
d. XDR
e. NFS
f. ANSI SQL

2.1.4 Interface Performance Requirements

This section is to be supplied.

2.2 Functional Specifications

2.2.1 Required Use of Existing Components

The File Server Building Block shall use COTS products and reuse existing components to provide the desired capabilities.

2.2.2 Functional Requirements Modifications

This section is to be supplied.

2.2.3 Performance Requirements Modifications

No performance-related modifications are required for the File Server Building Block.

2.2.4 Operational Requirements Modifications

No operational requirements modifications are required for the File Server Building Block.

3. Programmatic Requirements

3.1 Development Considerations

The File Server Building Block shall conform to the TBS development environment as specified in the TBS and to the standards listed in Section 2.1.3.

3.2 Portability

The File Server Building Block shall be POSIX compliant, and shall compile and execute on the ACE workstation platform.

3.3 Expandability

The File Server Building Block shall be expandable to allow additional storage capabilities and new types storage media.

3.4 Built-In Flexibility and Customization

The File Server Building Block shall parameterize the types and formats of files to be stored, processed, and delivered; the file access privileges; and all the parameters currently identified in the TPOCC file maintenance configuration file.

(Reference: TPOCC DDS for Release 10, Section C.9.3.5)

History Services (DS11)
Date of Specification: August 16, 1994

The development approach for the History Services Building Block is modify and then reuse the TPOCC history services.

1. Functional and Performance Overview

The History Services Building Block provides the ACE ground system with the capability to log, delog, and replay annotated spacecraft telemetry packets, command blocks, command echo blocks, and DSN monitor blocks. This building block also provides the capability to replay data from externally generated files. With a user request, the History Services Building Block opens a file and starts logging all of the objects of the requested data type to the file. When requested to stop logging the data, the History Services Building Block closes the file and records an entry in the database, identifying the starting and stopping time of the data in the file. Subsequent requests can be made to delog or replay the data from the file into the system. The delog function provides formatted (printable) output of the logged data items. The replay function plays the logged items back into the ground system for reprocessing.

Log files are managed using the services of the File Server Building Block. Log files are stored online until their expiration date. They are then stored on offline media from which they are available on request within 24 hours.

1.1 Functional Requirements Summary

1.1.1 The History Services Building Block shall process user requests to start logging any of the following:

a. ACE annotated spacecraft telemetry packets
b. ACE spacecraft command blocks
c. ACE command echo blocks
d. DSN monitor blocks

1.1.2 The History Services Building Block shall process user requests to suspend or stop the logging of any of the above data items.

1.1.3 The History Services Building Block shall process user requests to replay the data items from a previously created log file.

1.1.4 The History Services Building Block shall process user requests to delog the data items from a previously created log file.

1.1.5 The History Services Building Block shall be capable of replaying data from critical design review (CDR) replay and spacecraft command files.

1.2 Performance Requirements Summary

The History Services Building Block shall replay the requested data items at up to six times the rate at which they were received. For externally generated files, the receipt data rate is assumed to be the source data rate rather than the rate at which the files are received.

2. Specifications

2.1 Interface Specifications (Figure DS11-1)

The History Services Building Block shall use the ACE software backplane to interface with the following ACE ground system building blocks:

a. Packet Server
b. Command Echo Processor
c. DSN Monitor Block Processor
d. External Source of Command Blocks
e. Applications Services
f. File Server
g. Event Logger
h. System clock

2.1.1 Input

2.1.1.1 Packet Server Building Block: The History Services Building Block shall receive annotated telemetry packets in an ACE-specified CCSDS format from the Packet Server Building Block through the ACE software backplane.

[References: *ACE Command and Data Handling Component Specifications* (Preliminary), January 28, 1994, and TPOCC DDS for Release 10, Section C.7.4.1.3]

2.1.1.2 Command Echo Processor Building Block: The History Services Building Block shall receive command echo messages from the Command Echo Processor Building Block through the ACE software backplane.

2.1.1.3 DSN Monitor Block Processor Building Block: The History Services Building Block shall receive DSN monitor blocks from the DSN Monitor Block Processor Building Block through the ACE software backplane.

2.1.1.4 External Source of Command Blocks: The History Services Building Block shall receive ACE spacecraft command blocks from the external source of command blocks through FTP and the TCP/IP.

2.1.1.5 File Server Building Block: The History Services Building Block shall retrieve previously completed log files from the File Server Building Block for purposes of delogging or replaying these files and files received from external sources, such as CDR replay files from the Jet Propulsion Laboratory (JPL).

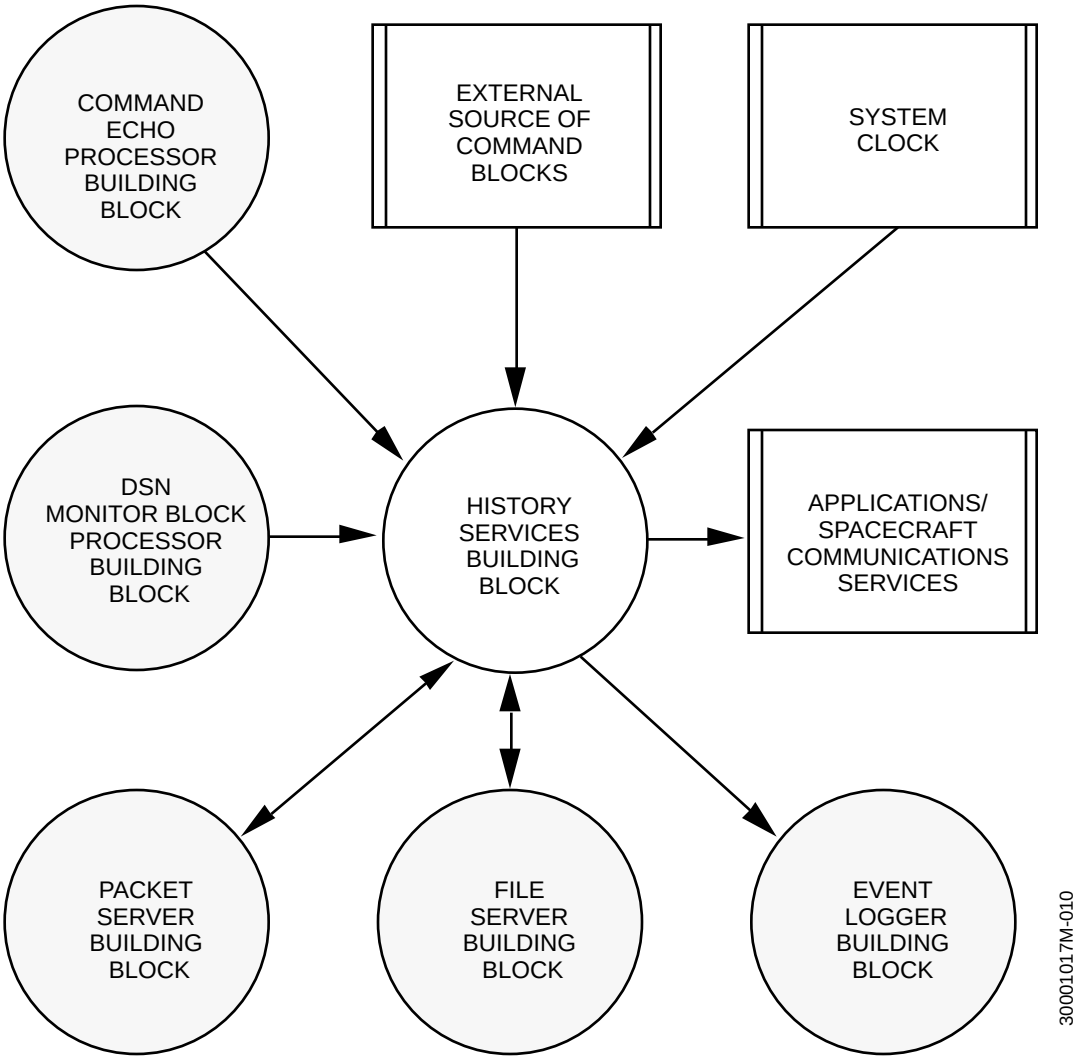

Figure DS11–1. History Services Building Block Context

2.1.1.6 System Clock: The History Services Building Block shall receive synchronized time from the system clock.

2.1.2 Output

2.1.2.1 Packet Server Building Block: The History Services Building Block shall provide annotated telemetry packets in an ACE-specified CCSDS format to the Packet Server Building Block through the ACE software backplane.

[Reference: *ACE Command and Data Handling Component Specifications* (Preliminary), January 28, 1994]

2.1.2.2 Other ACE Building Blocks

The History Services Building Block shall provide command echo, DSN monitor, and ACE spacecraft command blocks to other ACE building blocks through the ACE software backplane.

2.1.2.3 File Server Building Block: The History Services Building Block shall send completed log files to the File Server Building Block through the ACE software backplane for storage and management.

2.1.2.4 Event Logger Building Block: The History Services Building Block shall send event messages to the Event Logging Building Block through the ACE software backplane when a new logging, delogging, or replay session has begun or has been terminated, or when a log, delog, or replay request cannot be satisfied.

2.1.2.5 Spacecraft Communications Services: The History Services Building Block shall replay the contents of the CDR replay files to Spacecraft Communications Services via the software backplane.

2.1.3 Standards

The History Services Building Block shall comply with the following standards:

a. TCP/IP
b. POSIX
c. ANSI C

2.1.4 Interface Performance Requirements

The History Services Building Block shall be capable of processing

(10 packets per second x 6) + one command per second + one command echo per second + 1 DSN monitor block per second = 63 entries per second

2.2 Functional Specifications

2.2.1 Required Use of Existing Components

The History Services Building Block shall be based on the TPOCC history services.

(Reference TPOCC DDS for Release 10, Section 3.9)

2.2.2 Functional Requirements Modifications

The History Services Building Block shall provide the following modified functions:

a. Logging, replaying, and delogging annotated CCSDS telemetry packets, command blocks, DSN monitor blocks, and command echo messages
b. Logging, replaying, and delogging CCSDS telemetry packets that may have quality information attached to the packets
c. Replaying the contents of files containing spacecraft commands and CDR replays

2.2.3 Performance Requirements Modifications

The History Services Building Block shall be capable of replaying logged CCSDS packets at up to six times the received rate. For externally generated files, the receipt data rate is assumed to be the source data rate rather than the rate at which the files are received.

3. Programmatic Requirements

3.1 Development Considerations

The History Services Building Block shall conform to the ACE development environment as specified in the TBS and to the standards listed in Section 2.1.3.

3.2 Portability

The History Services Building Block shall be POSIX and ANSI C compliant, and shall compile and execute on the ACE workstation platform.

3.3 Expandability

The History Services Building Block shall be expandable to allow additional data types to be logged, delogged, and replayed.

3.4 Built-In Flexibility and Customization

The History Services Building Block shall use all parameters currently identified in the TPOCC block/frame configuration files.

(Reference: TPOCC DDS for Release 10, Sections C.9.3.3 and C.9.3.4)

Ingest Project Database (DS12a)
Date of Specification: August 16, 1994

The Ingest PDB Building Block is a new component.

1. Functional and Performance Overview

The Ingest PDB Building Block provides the ACE ground system with the capability of loading the ACE ground system database from the mission-specific PDB. The PDB defines the format of the ACE spacecraft telemetry and commands. Once the ACE database has been loaded, the operational data files that control decommutation can be derived and the applications performing spacecraft commanding can access it.

1.1 Functional Requirements Summary

The Ingest PDB Building Block shall read in the ACE-specific PDB files, transform the PDB records into the format required by the ACE database, and insert the transformed records into the ACE database through the Database Building Block and the ACE software backplane.

1.2 Performance Requirements Summary

The Ingest PDB Building Block shall be capable of ingesting the ACE PDB in TBS minutes.

2. Specifications

2.1 Interface Specifications (Figure DS12a-1)

The Ingest PDB Building Block shall use the ACE software backplane to interface with the following ACE ground system building blocks:

a. File Server
b. Database
c. User Services

2.1.1 Input

2.1.1.1 Project Database: The Ingest PDB Building Block shall access the PDB files through the capabilities of the File Server Building Block.

2.1.1.2 User Services: The Ingest PDB Building Block shall process directives from TBS to ingest a new PDB.

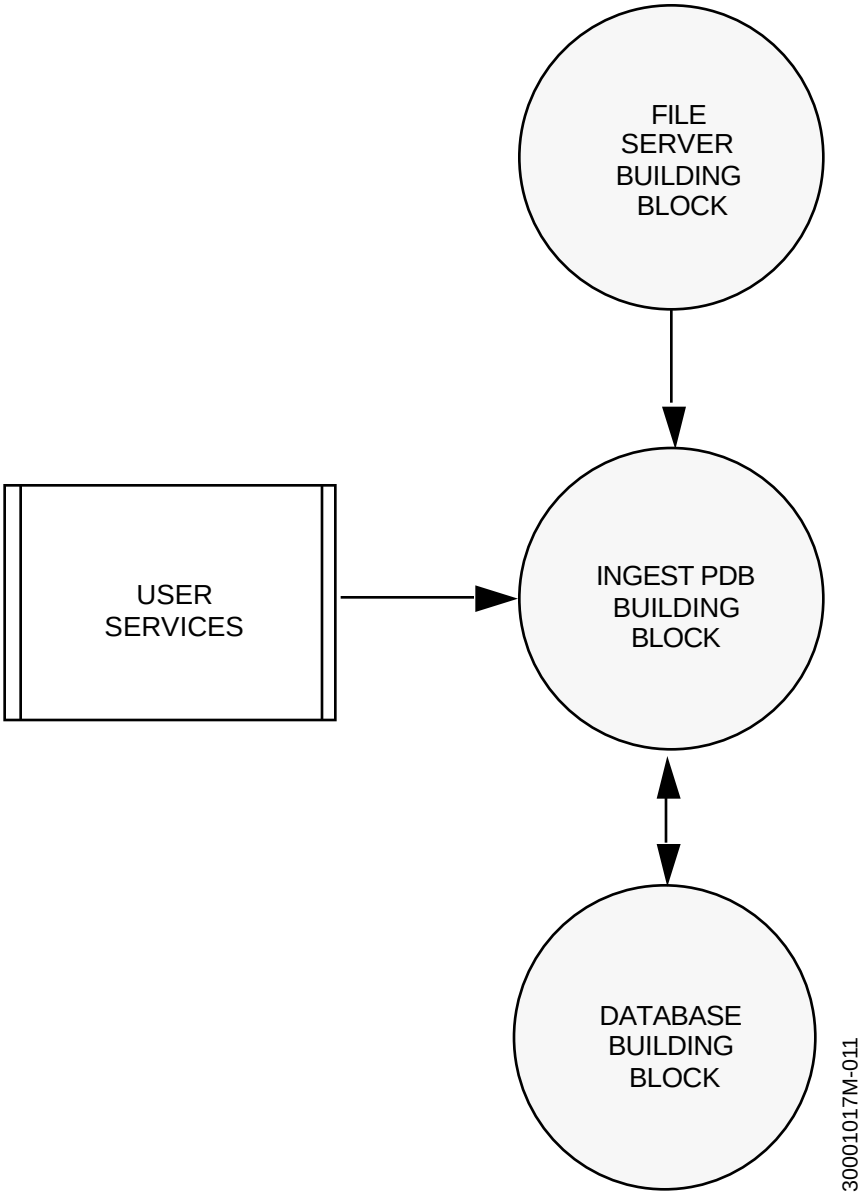

Figure DS12a–1. Ingest PDB Building Block Context

2.1.2 Output

2.1.2.1 Database Building Block: The Ingest PDB Building Block shall provide to the Database Building Block database entries that define the

a. Telemetry mnemonic, location, length, and data format for each parameter in the ACE spacecraft telemetry
b. Equation processing required on the telemetry parameters
c. Command mnemonic, format, and criticality of each ACE spacecraft command

2.1.3 Standards

The Ingest PDB Building Block shall comply with the following standards:

a. TCP/IP
b. POSIX
c. ANSI C
d. NFS
e. XDR
f. SQL (ANSI X3.135/X3.168)

2.1.4 Interface Performance Requirements

The Ingest PDB Building Block shall be capable of ingesting the ACE PDB in TBS minutes.

2.2 Functional Specifications

2.2.1 Functional Requirements Modifications

The Ingest PDB Building Block shall provide the following functions:

a. Read each ACE PDB file
b. Transform file records into format required by ACE database
c. Insert entries into ACE database required to decommutate the ACE telemetry
d. Insert entries into ACE database required to command the ACE spacecraft

2.2.2 Performance Requirements Modifications

The Ingest PDB Building Block shall be capable of ingesting the ACE PDB in TBS minutes.

2.2.3 Operational Requirements Modifications

2.2.3.1 Computer-Human Interface Requirements: The Ingest PDB Building Block has no computer-human interface requirements.

2.2.3.2 Training Requirements: The Ingest PDB Building Block has no training requirements.

2.2.3.3 Maintenance Requirements: The Ingest PDB Building Block has no maintenance requirements.

2.2.3.4 Safety Requirements: The Ingest PDB Building Block shall comply with AIS security level 2.

2.2.3.5 Security Requirements: Access to the Ingest PDB functionality shall be limited to authorized personnel.

3. Programmatic Requirements

3.1 Development Considerations

The Ingest PDB Building Block shall conform to the TBS development environment as specified in the TBS and to the standards listed in Section 2.1.3.

3.2 Portability

The Ingest PDB Building Block shall be POSIX compliant, and shall compile and execute on the ACE workstation platform.

3.3 Expandability

The Ingest PDB Building Block shall be expandable to handle different DBMSs and new PDB file formats.

3.4 Built-In Flexibility and Customization

The Ingest PDB Building Block shall parameterize PDB file formats.

Extract Project Database (DS12b)
Date of Specification: August 16, 1994

The Extract PDB Building Block is a combination of a new component and reuse of an existing component with modifications.

1. Functional and Performance Overview

The Extract PDB Building Block provides the ACE ground system with the following capabilities:

a. Creating from ACE database entries the operational data files required by the Decommutation Building Block
b. Reconstructing the PDB from the ACE ground system database content

1.1 Functional Requirements Summary

The Extract PDB Building Block shall build the operational Decommutation Building Block files from the ACE database contents and reconstruct the ACE PDB for the ACE database contents.

1.2 Performance Requirements Summary

The Extract PDB Building Block shall create the operational Decommutation Building Block files in TBS minutes and reconstruct the PDB in TBS minutes.

2. Specifications

2.1 Interface Specifications (Figure DS12b-1)

The Extract PDB Building Block shall use the capabilities of the ACE software backplane to interface with the following ACE ground system building blocks:

a. File Server
b. Database
c. User Services

2.1.1 Input

2.1.1.1 Database Building Block: The Extract PDB Building Block shall receive from the Database Building Block database entries that define

a. Telemetry mnemonic, location, length, and data format for each parameter in the ACE spacecraft telemetry
b. Equation processing required on the telemetry parameters
c. Command mnemonic, format, and criticality of each ACE spacecraft command

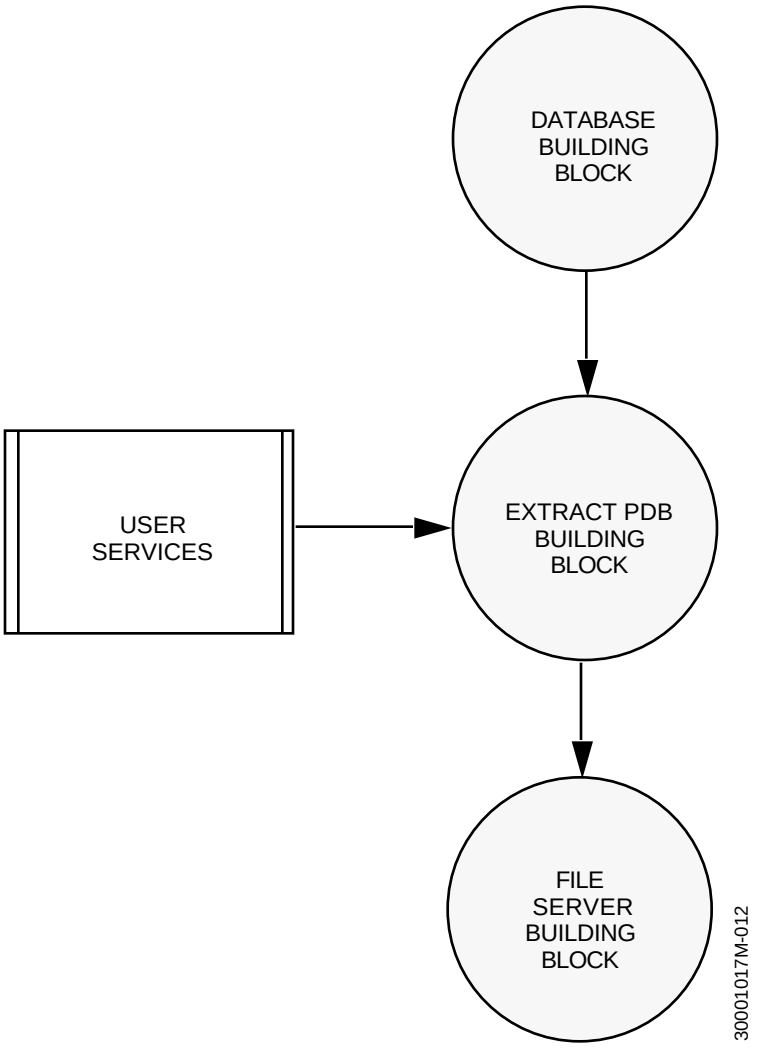

Figure DS12b–1. Extract PDB Building Block Context

2.1.1.2 User Services: The Extract PDB Building Block shall process directives from TBS to extract the PDB.

2.1.2 Output

The Extract PDB Building Block shall provide the following to the File Server Building Block:

a. Operational Decommutation Building Block files
b. Files containing the reconstructed ACE PDB

2.1.3 Standards

The Extract PDB Building Block shall comply with the following standards:

a. TCP/IP
b. POSIX
c. ANSI C
d. NFS
e. XDR
f. SQL (ANSI X3.135/X3.168)

2.1.4 Interface Performance Requirements

The Extract PDB Building Block has no interface performance requirements.

2.2 Functional Specifications

2.2.1 Required Use of Existing Components

The Extract PDB Building Block shall be based on the TPOCC tasks developed for the Tropical Rainfall Measuring Mission to build the operational files used by the decommutation processes from the ground system database.

2.2.2 Functional Requirements Modifications

The Extract PDB Building Block shall be capable of

a. Transforming the telemetry-defining database entries into the operational data files required by the Decommutation Building Block
b. Reconstructing the ACE PDB from the contents of the ACE database

2.2.3 Performance Requirements Modifications

2.2.3.1 The Extract PDB Building Block shall be capable of creating the open database from the ACE database in TBS minutes.

2.2.3.2 The Extract PDB shall be able to reconstruct the PDB from the contents of the ACE database in TBS minutes.

2.2.4 Operational Requirements Modifications

2.2.4.1 Computer-Human Interface Requirements: The Extract PDB Building Block has no additional computer-human interface requirements.

2.2.4.2 Training Requirements: The Extract PDB Building Block has no additional training requirements.

2.2.4.3 Maintenance Requirements: The Extract PDB Building Block has no additional maintenance requirements.

2.2.4.4 Safety Requirements: The Extract PDB Building Block has no additional safety requirements.

2.2.4.5 Security Requirements: The Extract PDB Building Block shall comply with AIS Security Level 2.

3. Programmatic Requirements

3.1 Development Considerations

The Extract PDB Building Block shall conform to the TBS development environment as specified in the TBS and to the standards listed in Section 2.1.3.

3.2 Portability

The Extract PDB Building Block shall be POSIX compliant, and shall compile and execute on the ACE workstation platform.

3.3 Expandability

The Extract PDB Building Block shall be expandable to handle different DBMSs and new PDB file formats.

3.4 Built-In Flexibility and Customization

The Extract PDB Building Block shall parameterize PDB file formats.

Database (DS13)
Date of Specification: July 1, 1994

Functionality for the Database Building Block will be provided by a DBMS, such as ORACLE. Reuse schema and procedures will come from TPOCC and Pacor II. Modify and enhance as necessary to meet all database needs.

1. Functional and Performance Overview

1.1 Functional Requirements Summary

The Database Building Block shall provide storage and retrieval for data values using relational database technology. The database shall protect data from unauthorized disclosure, update, and deletion. The database shall provide the capability to format data into reports, and shall have the capability to export data to files in industry standard formats that can be ingested by other software.

1.2 Performance Requirements Summary

The Database Building Block shall respond to queries in TBS seconds. The database shall complete database updates with an average size of TBS bytes in TBS seconds.

2. Specifications

2.1 Interface Specifications (Figure DS13-1)

2.1.1 Input

The Database Building Block shall receive updates from the PDPS, Real-Time Subset Generation, High-Speed Offline Subset, Ingest PDB, Applications Services, User Services, Spacecraft Communications Services, and Simulation Services Building Blocks.

2.1.2 Output

The Database Building Block shall output results to queries of the information contained in its tables. The database shall generate reports based on stored specifications (creation of specifications is the responsibility of User Services). The database shall generate export files in industry standard formats, such as American Standard Code for Information Interchange (ASCII) text with user-selected delimiters.

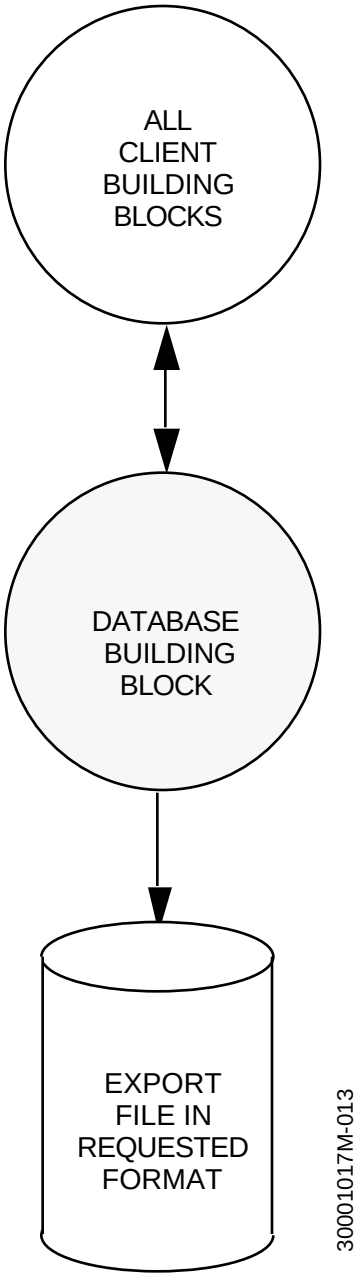

Figure DS13–1. Database Building Block Context

2.1.3 Standards

The Database Building Block shall comply with the following standards:

a. ANSI SQL
b. ANSI-embedded SQL
c. TCP/IP

2.1.4 Interface Performance Requirements

Other building blocks shall interface with the Database Building Block through the services of the software backplane.

2.2 Specifications

2.2.1 Required Use of Existing Components

The Database Building Block shall reuse database schema from the Pacor II database, as well as the PDPS database, to support the PDPS, quality analysis workstation subsystem (QAWS), and Pacor II information and control subsystem (PICS) functionality reused within the ACE Mission Operations Center (MOC) and database schema from TPOCC to support trending analysis and other TBS building blocks.

2.2.2 Functional Requirements Modifications

The Database Building Block shall provide schema to support the ingest and extraction of the PDB, as well as queries against this information. The database shall provide access routines to new tables as needed to support the modified storage needs of other building blocks. New access routines shall conform to the standards set forth in Section 2.1.3 to ensure portability across database engines.

2.2.3 Data Management Requirements Modifications

2.2.3.1 Data Management Architecture

2.2.3.1.1 Data Model: The data model for the Pacor II and PDPS databases is found in the Pacor II DDS, May/August 1993. The data model for the trending analysis is found in TBS.

2.2.3.1.2 Data Structure Requirements: The database shall merge the data structures needed to support all of the building blocks into a single database. The database shall incorporate new tables as needed to support the modified storage needs of other building blocks.

2.2.3.2 Operational Environment Requirements

2.2.3.2.1 Initialization Requirements: The database shall be initialized with the necessary parameter values to support the ACE mission. These parameters include ACE LZP setup information, TBS.

2.2.3.2.2 Availability Requirements: The database shall be available throughout each staffed shift. Database maintenance will be performed during the unstaffed shifts.

2.2.4 Operational Requirements Modifications

2.2.4.1 Computer-Human Interface Requirements: The computer-human interface with the database will be handled through User Services.

2.2.4.2 Training Requirements: A designated database administrator will need to be trained in the administration of the COTS package utilities, as well as the structure of the database.

2.2.4.3 Maintenance Requirements: The database shall be backed up to offline media every TBS days. Information more than TBS days old shall be removed from the online database following a successful backup.

2.2.4.4 Security Requirements: The database shall comply with NASA AIS Level 1 security requirements to prevent unauthorized access, modification, and destruction of the data contained in the database.

3. Programmatic Requirements

3.1 Development Considerations

The Database Building Block shall be developed using Oracle's Relational Database Management System to facilitate reuse of existing schema, access procedures, and utilities.

3.2 Portability

The database schema shall be modularized to allow the transfer of building blocks supported by the database to other facilities, without requiring the transfer of the entire database.

3.3 Expandability

The Database Building Block has no expandability requirements.

3.4 Built-In Flexibility and Customization

The database schema shall be modularized to allow it to be scaled to meet the specific building block needs of future missions.

Real-Time Subset Generation (DS14)
Date of Specification: August 10, 1994

The development approach for the Real-Time Subset Generation Building Block is to modify the existing TPOCC Real-Time Subset task.

Requirements: DS-020, DS-023, and DS-033

1. Functional and Performance Overview

The Real-Time Subset Generation Building Block will produce binary subset data and subset summaries from requested decommutated telemetry data based on subset definitions. The Real-Time Subset Generation Building Block will receive setup directives; interface with the Data Server Building Block, State Manager Function, Event Logger Building Block, database server, and system clock; and send setup responses and event messages.

1.1 Functional Requirements Summary

The Real-Time Subset Generation Building Block shall

a. Interface with the following:
   1. Data Server Building Block (Reference: TPOCC DDS Release 10, Section C.11.1.1)
   2. State Manager Function through the software backplane
   3. Event Logger Building Block (Reference: TPOCC DDS Release 10, Section C.10.1.4)
   4. System clock
   5. Database server
b. Receive decommutated engineering data from the data server
c. Provide binary subset data and subset summaries
d. Send formatted event messages consisting of an index to the event type, a criticality indicator, and the event message text (the receipt time should be stored in the first word of the event message text)
e. Receive directives from the State Manager Function
f. Send directive responses to the State Manager Function
g. Retrieve the time from the system clock on demand
h. Retrieve subset definitions from the database server through the software backplane
i. Receive DSN Monitor Block parameters from the Data Server Building Block.

1.2 Performance Requirements Summary

1.2.1 The Real-Time Subset Generation Building Block shall receive decommutated data from the data server up to a maximum file size of TBS.

1.2.2 The Real-Time Subset Generation Building Block shall provide subsets based on up to 2000 parameters per subset.

1.2.3 The Real-Time Subset Generation Building Block shall generate the subsets within TBS time after receipt of the data.

2. Specifications

2.1 Interface Specifications (Figure DS14-1)

2.1.1 The Real-Time Subset Generation Building Block shall interface with the Data Server and Event Logger Building Blocks. The interface shall use TCP/IP socket connections in a client/server environment.

(Reference: TPOCC DDS Release 10, Section C.10.1.4)

2.1.2 The Real-Time Subset Generation Building Block shall interface with the system clock to retrieve the time.

2.1.3 The Real-Time Subset Generation Building Block shall interface with the State Manager Function through the software backplane.

2.1.4 The Real-Time Subset Generation Building Block shall interface with the database server through the software backplane.

2.1.1 Input

2.1.1.1 Data Server Building Block: The Real-Time Subset Generation Building Block shall receive requested decommutated telemetry data and DSN Monitor Block parameters from the Data Server Building Block.

(Reference: TPOCC DDS for Release 10, Section C.11.1.1)

2.1.1.2 State Manager Function: The Real-Time Subset Generation Building Block shall receive setup directives from the State Manager Function through the software backplane.

2.1.1.3 System Clock: The Real-Time Subset Generation Building Block shall retrieve the time from the synchronized system clock.

2.1.1.4 Database Server: The Real-Time Subset Generation Building Block shall receive subset definitions from the database server.

2.1.2 Output

2.1.2.1 Data Server Building Block: The Real-Time Subset Generation Building Block shall send specifications for requested data to the data server.

(Reference: TPOCC DDS for Release 10, Section C.11.1.1)

2.1.2.2 State Manager Function: The Real-Time Subset Generation Building Block shall send directive status messages to the State Manager Function through the software backplane.

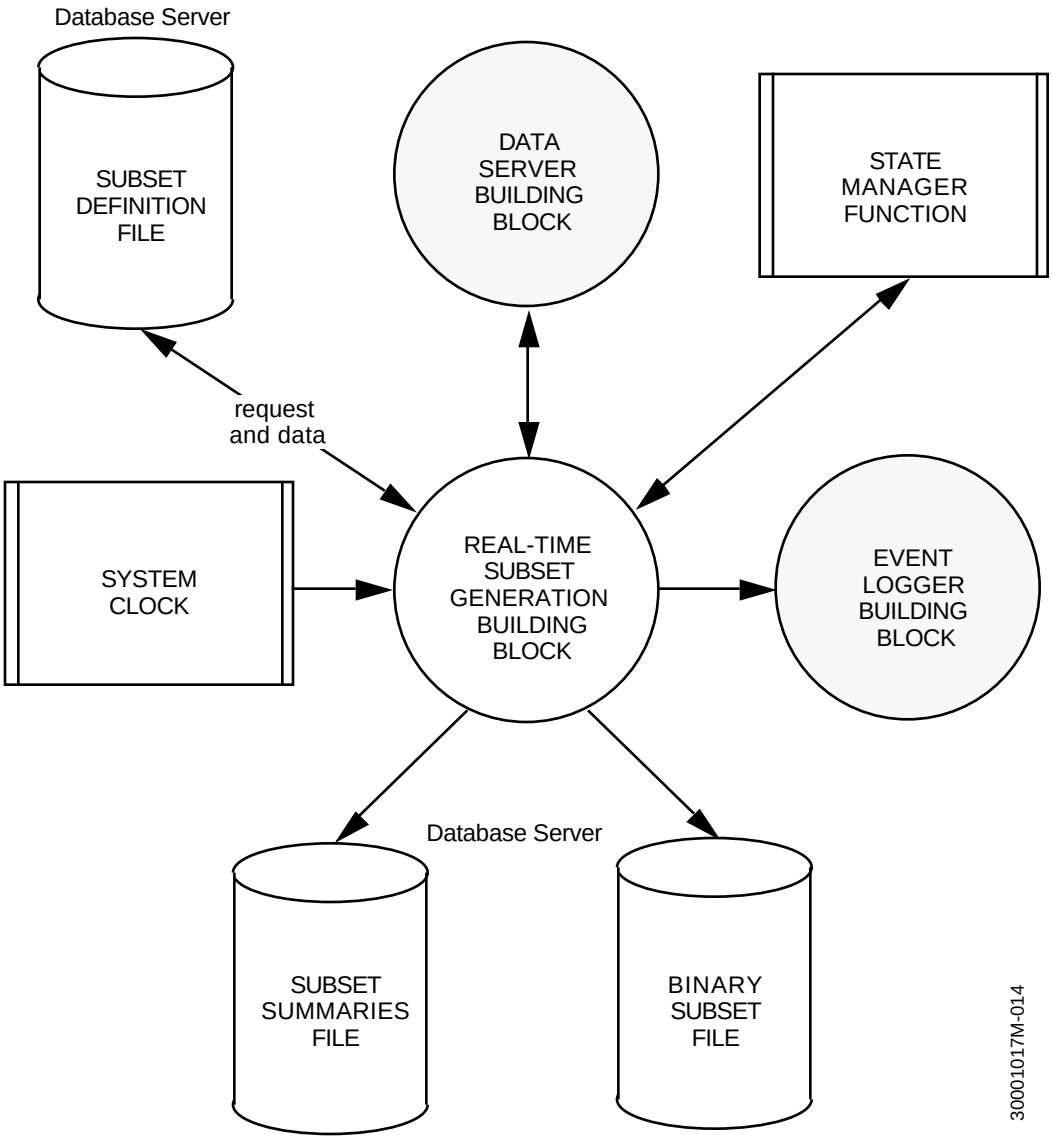

Figure DS14–1. Real-Time Subset Generation Building Block Context

2.1.2.3 Event Logger Building Block: The Real-Time Subset Generation Building Block shall send formatted event messages consisting of an index to the event type, a criticality indicator, and the event message text to the Event Logger Building Block. The receipt time should be stored in the first word of the event message text.

(Reference: TPOCC DDS for Release 10, Section C.10.1.4)

2.1.2.4 Database Server: The Real-Time Subset Generation Building Block shall send binary subsets and subset summaries to the database server.

2.1.3 Standards

The Real-Time Subset Generation Building Block shall comply with the following standards:

a. TCP/IP
b. POSIX.2/4
c. ANSI C
d. NFS
e. XDR
f. IEEE Floating Point
g. ANSI SQL

2.2 Functional Specifications

2.2.1 Required Use of Existing Components

The Real-Time Subset Generation Building Block shall reuse the TPOCC Real-Time Subset task.

(Reference: TBS)

2.2.2 Functional Requirements Modifications

The Real-Time Subset Generation Building Block shall be totally configurable by subset definition information and by State Manager Function directives.

3. Programmatic Requirements

3.1 Development Considerations

The Real-Time Subset Generation Building Block shall conform to the ACE development environment and to the standards listed in Section 2.1.3.

3.2 Portability

The Real-Time Subset Generation Building Block shall be POSIX and ANSI C compliant, and shall compile and execute on both the ACE real-time and workstation platforms.

3.3 Expandability

The Real-Time Subset Generation Building Block shall be expandable to allow subsets of information to TBS parameters.

3.4 Built-In Flexibility and Customization

The following types of data shall be parameterized:

a. Number of parameters per subset
b. Unique parameters to be included in each subset (by mnemonic)
c. Type of parameter (raw, converted, or both)
d. Telemetry parameter definitions
e. Calibration definitions
f. Limits definitions

Packet Server (DS15)
Date of Specification: August 10, 1994

The development approach for the Packet Server Building Block is to modify the existing TPOCC PACKET_EXTRACTOR task.

Requirements: DS-001, DS-005, DS-008, DS-020, and DS-023

1. Functional and Performance Overview

The Packet Server Building Block will perform functions similar to the TPOCC telemetry packet extractor. The Packet Server Building Block will receive real-time and playback data (consisting of reassembled packets and quality annotations) from Spacecraft Communications Services; extract and format selected data; and route selected packets to its clients (internal and external).

(Reference: TPOCC DDS for Release 10, Section 2.7.4)

1.1 Functional Requirements Summary

1.1.1 The Packet Server Building Block shall provide similar functional capabilities as the TPOCC PACKET_EXTRACTOR task. However, this building block shall not interface with the TNIF.

1.1.2 The Packet Server Building Block shall

a. Receive real-time and playback telemetry consisting of reassembled packets
b. Log data to the History Services Building Block as required before processing
c. Receive TBS data quality annotations in TBS format
d. Validate received packets
e. Reformat quality annotations as required by client building blocks for use by other Data Server Building Blocks
f. Accept directives from the State Manager Function through the software backplane
g. Generate events in accordance with the current TPOCC packet extractor
h. Transfer filtered packets to client building blocks based on client-specified parameters
i. Request and receive configuration and packet definition information from the database server

1.2 Performance Requirements Summary

The Packet Server Building Block shall

a. Ingest data at up to an average rate of 76 kbps
b. Ingest replays of stored data at up to six times the original data rate (76 x 6 kbps = 456 kbps)
c. Receive, process, and forward real-time data within 1 second of receipt

2. Specifications

2.1 Interface Specifications (Figure DS15-1)

2.1.1 The Packet Server Building Block shall interface with the following building blocks, using TCP/IP socket connections in a client/server environment:

a. Spacecraft Communications Services
b. Telemetry Decommutation Building Block
c. Other building blocks that process telemetry packets
d. Event Logger Building Block (Reference: TPOCC DDS for Release 10, Section C 10.1.4)
e. History Services Building Block (Reference: TPOCC DDS for Release 10, Section C.9)

2.1.2 The Packet Server Building Block shall interface with the State Manager Function through the software backplane.

2.1.3 The Packet Server Building Block shall interface with the database server through the software backplane.

2.1.1 Input

2.1.1.1 Spacecraft Communications Services: The Packet Server Building Block shall receive the data stream consisting of annotated CCSDS real-time packets from Spacecraft Communications Services through TCP/IP socket connection.

2.1.1.2 State Manager Function: The Packet Server Building Block shall receive the following directives and control commands from the State Manager Function through the software backplane:

a. PASS_INIT (initialize global variables)
b. ACQUIRE_ON (connect to Spacecraft Communications Services)
c. ACQUIRE_OFF (disconnect from Spacecraft Communications Services)
d. PASS_OFF (terminates packet service for all packet clients)

2.1.1.3 Telemetry Decommutation and Other Client Building Blocks: The Packet Server Building Block shall receive the following client messages from the Telemetry Decommutation and other client building blocks that process telemetry packets:

a. Service activation
b. Filter table
c. Service deactivation
d. Disconnect

(Reference: TPOCC DDS for Release 10, Section C.7.4.1.3)

2.1.1.4 History Services Building Block: The Packet Server Building Block shall receive replay data at a maximum rate of 456 kbps from the History Services Building Block through a TCP/IP socket connection.

(Reference: TPOCC DDS for Release 10, Section C.9)

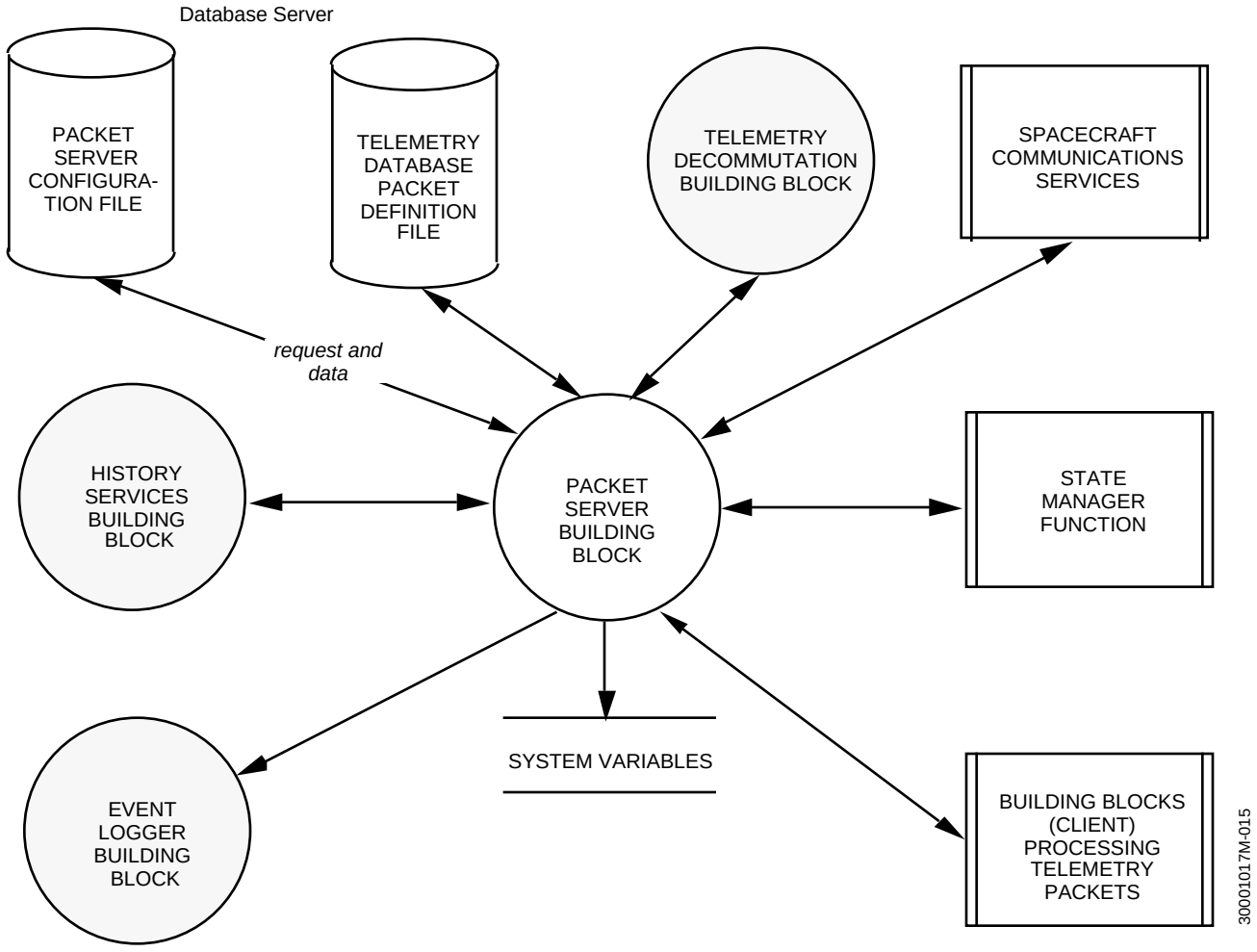

Figure DS15–1. Packet Server Building Block Context

2.1.1.5 Database Server: The Packet Server Building Block shall receive configuration and packet definition information from the database server through the software backplane.

2.1.2 Output

2.1.2.1 State Manager Function: The Packet Server Building Block shall send directive status messages to the State Manager Function through the software backplane.

2.1.2.2 Telemetry Decommutation and Other Client Building Blocks: The Packet Server Building Block shall send server (telemetry packets in an ACE-specified CCSDS format) and other messages to the Telemetry Decommutation and other client building blocks that request telemetry packets.

[References: *ACE Command and Data Handling Component Specifications* (Preliminary), January 28, 1994, and TPOCC DDS for Release 10, Section C.7.4.1.3]

2.1.2.3 Event Logger Building Block: The Packet Server Building Block shall send formatted event messages consisting of an index to the event type, a criticality indicator, and the event message text to the Event Logger Building Block. The receipt time should be stored in the first word of the event message text.

(Reference: TPOCC DDS for Release 10, Section C.10.1.4)

2.1.2.4 History Services Building Block: The Packet Server Building Block shall send annotated real-time and playback packets to the History Services Building Block through a TCP/IP socket connection.

(Reference: TPOCC DDS for Release 10, Section C.9)

2.1.1.5 Database Server: The Packet Server Building Block shall send requests for configuration and packet definition information to the database server through the software backplane.

2.1.3 Standards

The Packet Server Building Block shall comply with the following standards:

a. TCP/IP
b. POSIX.2/4
c. ANSI C
d. NFS
e. XDR
f. Packet telemetry per CCSDS 102.0-B-2
g. ANSI SQL

2.1.4 Interface Performance Requirements

2.1.4.1 Spacecraft Communications Service: The Packet Server Building Block interface to Spacecraft Communications Service shall provide the capability to receive data at an average rate

of 76 kbps and receive replays of stored data at up to six times the original data rate (76 x 6 kbps = 456 kbps).

2.1.4.2 State Manager Function: The Packet Server Building Block shall interface with the State Manager Function through the software backplane.

2.1.4.3 Event Logger Building Block: The Packet Server Building Block interface to the Event Logger Building Block shall perform as specified in TPOCC DDS for Release 10, Sections 3.10.2 and C.10.

2.1.4.4 History Services Building Block: The Packet Server Building Block interface to the History Services Building Block shall provide the capability to send and receive data at up to six times the original rate (76 x 6 kbps = 456 kbps).

2.2 Functional Specifications

2.2.1 Required Use of Existing Components

The Packet Server Building Block shall be based on the TPOCC PACKET_EXTRACTOR task.

(Reference: TPOCC DDS for Release 10, Section 3.7.3.2)

2.2.2 Functional Requirements Modifications

The Packet Server Building Block shall

a. Be modified to receive telemetry data from Spacecraft Communications Services through the TCP/IP
b. Not use the TNIF
c. Not process frame-level CCSDS data [virtual channel data unit (VCDU) or transfer frames]
d. Process telemetry packet formats as specified in the *ACE Command and Data Handling Component Specifications* (Preliminary), January 28, 1994
e. Reformat data quality information received with received telemetry
f. Provide annotated packets to the History Services Building Block for logging

2.2.3 Data Management Requirements Modifications

Data management requirements modifications are not applicable for this building block.

2.2.4 Performance Requirements Modifications

The Packet Server Building Block shall be capable of sending and receiving data at up to six times the original rate (76 x 6 kbps = 456 kbps) and of receiving, processing, and forwarding real-time data within 1 second of receipt.

2.2.5 Operational Requirements Modifications

Any operational requirements will be provided by User Services.

3. Programmatic Requirements

3.1 Development Considerations

The Packet Server Building Block shall conform to the ACE development environment as specified in TBS and to the standards listed in Section 2.1.3.

3.2 Portability

The Packet Server Building Block shall be POSIX and ANSI C compliant, and shall compile and execute on both the ACE real-time and workstation platforms.

3.3 Expandability

Currently, no expandability requirements are identified for this building block.

3.4 Built-In Flexibility and Customization

The following types of data shall be parameterized:

a. Description of received CCSDS packets
b. Description of data quality annotations
c. Description of overall data stream (for example, relative location and frequency of packets versus annotations)
d. Size of telemetry packet buffer table
e. Size of telemetry packet client table
f. Size of telemetry packet length table
g. Size of telemetry packet cyclic redundancy code (CRC) table
h. Size of telemetry packet parts table
i. Size of telemetry packet extractor configuration file
j. Size of records in the telemetry packet database file

Data Distribution (DS19)
Date of Specification: August 16, 1994

The development approach for the Data Distribution Building Block is to reuse the existing design.

1. Functional and Performance Overview

The Data Distribution Building Block provides the ACE ground system with the capabilities for cataloging and distributing ACE data products. It maintains a catalog of data sets stored on the ACE ground system, making it possible to quickly locate and retrieve data given its type and effective dates. The Data Distribution Building Block provides for automatic version control so that different versions of the same data can be identified and provides the capability to associate multiple files with a single data set. It also automatically distributes data products to destinations for which distribution specifications exist in the ACE database.

Other building blocks in the ACE ground system create and access files using the operating system and File Server Building Block capabilities. Applications requiring their files be cataloged create an SFDU header file referencing the files belonging to the data set. The SFDU header files are identified to the Data Distribution Building Block in a data availability notice. Distribution processes the request by extracting metadata from the SFDU file and creating entries in the catalog. The SFDU headers conform to standards that allow the use of a generic procedure for extracting metadata. The Data Distribution Building Block checks for the existence of previous versions of the same data and automatically assigns appropriate version numbers to new data sets.

When a data product is cataloged, the database is queried to determine if any requirements exist to distribute the product. If so, the product is transferred to the required destination(s) electronically or to the hardware that creates the distribution compact disk read-only memories (CD-ROMs).

This building block uses concepts and designs from the International Solar-Terrestrial Physics (ISTP) Central Data Handling Facility (CDHF) to provide a reusable component for ACE and for future missions. Applications program interfaces (APIs) currently being developed for the Flight Dynamics Division's (FDD's) interface to the Data Distribution Facility (DDF), also known as the DDFI, will be used to provide other building blocks with a means of communicating with the Data Distribution Building Block. The building block also uses the capabilities of CD-ROM creation hardware (such as that used by the DDF) to create the CD-ROMs that are distributed to ACE data product users.

(Reference: DDS for the ISTP CDHF, Section 5.1)

1.1 Functional Requirements Summary

The Data Distribution Building Block shall

a. Accept and service data availability messages from the other ACE building blocks to catalog data sets
b. Retrieve the detached SFDU headers referenced by the data availability message from the file server
c. Extract metadata from detached SFDU headers identifying the files in the data sets; this data shall include type of data in the files, names of physical files in data set, and effective coverage dates of files
d. Identify duplicate data sets and assign appropriate version numbers
e. Place an entry in the ACE database (through the Database Building Block) for each data set
f. Verify the existence of all data files identified in the SFDU header file
g. Place an entry in the ACE database for each data file that associates the file with a particular data set
h. Generate event messages identifying any data sets that fail the catalog process
i. Distribute data products to clients either electronically or through CD-ROM creation hardware and media
j. For data products distributed electronically, ensure that the distribution is completed successfully
k. For data products distributed through offline media, ensure successful delivery of data products to CD-ROM hardware
l. Provide the capability to create the data availability messages and detached SFDU headers for clients who lack this capability

1.2 Performance Requirements Summary

1.2.1 The Data Distribution Building Block shall be capable of cataloging the following files in the amounts indicated:

File Type	Maximum No. of Files
TBS	TBS

1.2.2 The Data Distribution Building Block shall be capable of distributing electronically the following files in the amounts indicated:

File Type	Maximum No. of Files
TBS	TBS

1.2.3 The Data Distribution Building Block shall be capable of distributing on CD-ROM the following files in the amounts indicated:

File Type	Maximum No. of Files
TBS	TBS

1.2.4 The Data Distribution Building Block shall be capable of distributing the following types of data within the indicated period of time:

Product Type	To MOC	To ASC
Production data	12 hours	9.5 days
Quick LZP	1.5 hours	4.5 hours
Quicklook	1.5 hours	4.5 hours

2. Specifications

2.1 Interface Specifications (Figure DS19-1)

The Data Distribution Building Block shall interface with other ACE ground system building blocks in a client/server environment using the capabilities of the ACE software backplane. It shall interface with the following building blocks:

a. Database Building Block
b. File Server Building Block
c. Any other ACE building blocks
d. CD-ROM creation hardware
e. ACE FTP clients
f. Event Logging Building Block

2.1.1 Input

2.1.1.1 Other ACE Building Blocks: The Data Distribution Building Block shall accept data availability notifications identifying the detached SFDU header files for data sets to be cataloged from the other ACE building blocks.

2.1.1.2 File Server Building Block: The Data Distribution Building Block shall interface with the File Server Building Block to

a. Retrieve detached SFDU headers in the project-specified standard format that identify the names and content of data files in the data set
b. Check the existence of data files referenced by the detached SFDU headers

2.1.1.3 Database Building Block: The Data Distribution Building Block shall retrieve the following from the Database Building Block through the ACE software backplane:

a. Identification of data files already cataloged
b. Distribution specifications identifying destinations and media for ACE data products requiring distribution

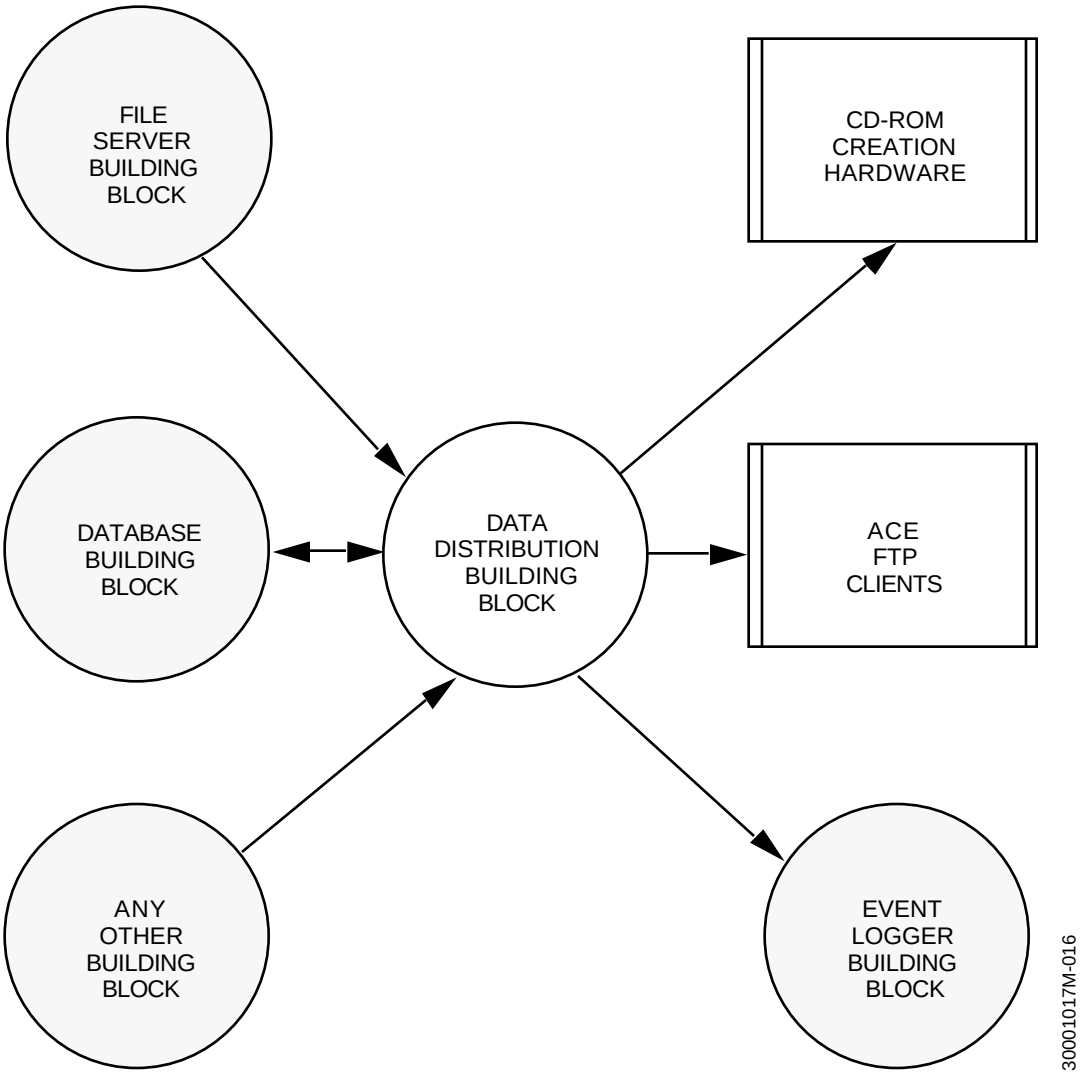

Figure DS19–1. Data Distribution Building Block Context

2.1.2 Output

2.1.2.1 Database Building Block: The Data Distribution Building Block shall output the following to the Database Building Block through the ACE software backplane:

a. SQL-formatted requests to insert an entry identifying the data set being cataloged
b. SQL-formatted requests to insert entries identifying each of the data files in the data set being cataloged
c. Queries for data product distribution specifications

2.1.2.2 Event Logging Building Block: The Data Distribution Building Block shall output the following event messages to the Event Logging Building Block:

a. Those identifying data sets that have been cataloged
b. Those identifying data sets that fail the catalog process
c. Those identifying distributions that it has been unable to complete

2.1.2.3 ACE FTP Clients: The Data Distribution Building Block shall output ACE data products electronically to FTP clients.

2.1.2.4 CD-ROM Creation Hardware: The Data Distribution Building Block shall transfer ACE data products to the CD-ROM creation hardware to create CD-ROMs for distribution to authorized clients.

2.1.3 Standards

The Data Distribution Building Block shall comply with the following standards:

a. TCP/IP
b. POSIX
c. ANSI C
d. ACE project-specific SFDU header format
e. SQL (ANSI X3.135/X3.168)

2.1.4 Interface Performance Requirements

The Data Distribution Building Block shall be capable of cataloging files at the rate of TBS data sets per day.

2.2 Functional Specifications

2.2.1 Required Use of Existing Components

The Data Distribution Building Block shall use the conceptual design of the ISTP CDHF Catalog Data task.

2.2.2 Functional Requirements Modifications

The functional characteristics of the ISTP CATALOG_DATA task shall be modified to

a. Adapt to any differences between the ISTP-standard SFDU header file format and that defined for the ACE project
b. Accept data availability messages and shutdown requests through the ACE software backplane rather than a VAX mailbox
c. Communicate event and error messages to the Event Logger Building Block (through the ACE software backplane) rather than the existing operator interface mailbox
d. Remove any functionality related specifically to orbit/attitude work and key parameter files
e. Add the capability to interface with the hardware that creates CD-ROMs
f. Add the capability to distribute data products to ACE clients through the FTP

The functional characteristics of the FDF DDFI software shall be modified to

a. Disable the DDF interface protocol because the ACE mission data distribution will be internal to the MOC and no response is necessary.
b. Allow APIs to be used by the data generation building blocks to communicate with the Data Distribution Building Blocks.

2.2.3 Performance Requirements Modifications

No performance related modifications are required for the Data Distribution Building Block.

2.2.4 Operational Requirements Modifications

No operational requirements modifications are required for the Data Distribution Building Block.

3. Programmatic Requirements

3.1 Development

The Data Distribution Building Block shall conform to the TBS development environment as specified in the TBS and to the standards listed in Section 2.1.3.

3.2 Portability

The Data Distribution Building Block shall be POSIX compliant, and shall compile and execute on the ACE workstation platform.

3.3 Expandability

The Data Distribution Building Block shall be expandable to allow the addition of capabilities for FTP clients to pull data products from the ACE online storage and to provide data products on other media, such as magnetic tapes.

3.4 Built-In Flexibility and Customization

The Data Distribution Building Block shall be configured by database entries and external files. The following items shall be parameterized:

a. Expected SFDU format
b. Delivery specifications including electronic formats, hard media formats, and routing information
c. Event message definitions
d. Delivery options for each file type

ACE TDM Processing (DS20)
Date of Specification: August 1, 1994

The ACE Time-Division Multiplexed (TDM) Processing Building Block is to be developed specifically for the ACE mission.

1. Functional and Performance Overview

1.1 Functional Requirements Summary

The ACE TDM Processing Building Block shall provide counts of complete and incomplete major frames and missing minor frames for a single pass.

1.2 Performance Requirements Summary

The ACE TDM Processing Building Block shall be capable of processing TBS minor frames within TBS minutes.

2. Specifications

2.1 Interface Specifications (Figure DS20-1)

The ACE TDM Processing Building Block shall interface with the File Server Building Block, the system clock, the Database Building Block, the Event Logger Building Block, and the State Manager Function.

2.1.1 Input

2.1.1.1 File Server Building Block: The ACE TDM Processing Building Block shall receive LZP products sorted by spacecraft time and packet source sequence count from the File Server Building Block.

2.1.1.2 System Clock: The ACE TDM Processing Building Block shall receive the current time from the synchronized system clock.

2.1.1.3 State Manager Function: The ACE TDM Processing Building Block shall receive “terminate ACE TDM processing” directives from the State Manager Function.

2.1.2 Output

2.1.2.1 Database Building Block: The ACE TDM Processing Building Block shall provide major and minor frame accounting statistics to the Database Building Block through the software backplane as described in Section 2.1.1.

2.1.2.2 Event Logger Building Block: The ACE TDM Processing Building Block shall not provide output to the Event Logger Building Block.

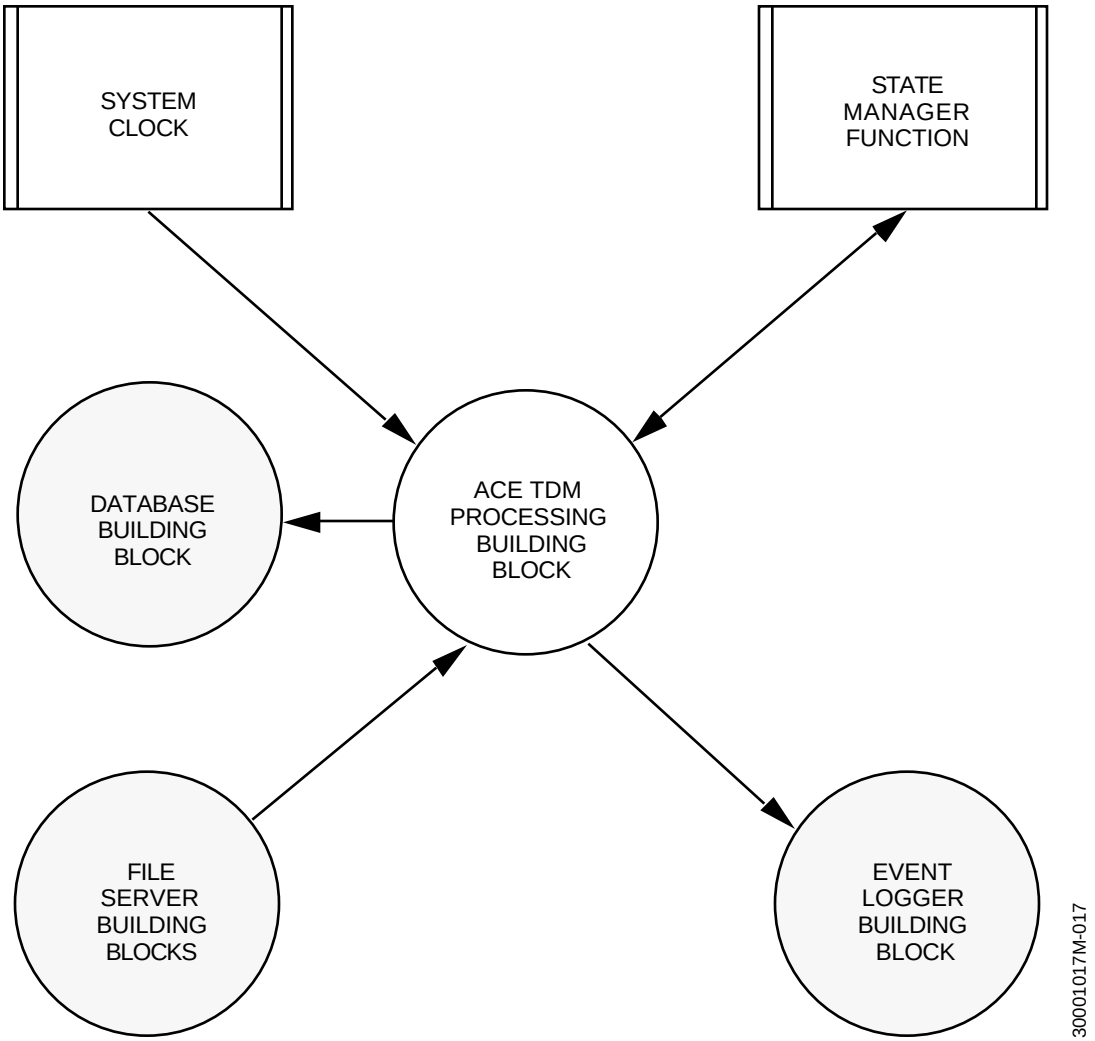

Figure DS20–1. ACE TDM Processing Building Block Context

2.1.2.3 State Manager Function: The ACE TDM Processing Building Block shall provide directive status messages to the State Manager Function.

2.1.3 Standards

The ACE TDM Processing Building Block shall comply with the following standards:

a. POSIX
b. ANSI SQL
c. NFS

2.1.4 Interface Performance Requirements

This section is to be supplied.

2.2 Functional Specifications

2.2.1 Functional Requirements Modifications

2.2.1.1 The ACE TDM Processing Building Block shall provide counts from a quick LZP product as follows:

a. Complete major frames
b. Incomplete major frames
c. Missing minor frames

2.2.1.2 The ACE TDM Processing Building Block shall provide a list of incomplete major frames as follows:

a. Major frame number
b. Starting and ending VCDU number for the major frame
c. Starting and ending packet source sequence counter for the major frame
d. Missing minor frame number(s)
e. Starting and ending VCDU number of each missing minor frame
f. Packet source sequence counter of each missing minor frame

2.2.2 Performance Requirements

The ACE TDM Processing Building Block shall process TBS minor frames within TBS minutes.

3. Programmatic Requirements

3.1 Development Considerations

The ACE TDM Building Block shall be developed to comply with the ACE development environment as described in TBS.

3.2 Portability

The ACE TDM Processing Building Block shall compile and execute on any platform that supports POSIX and ANSI C standards.

3.3 Expandability

The ACE TDM Processing Building Block is being developed for the ACE mission.

3.4 Built-In Flexibility and Customization

The ACE TDM Processing Building Block is being developed for the ACE mission.

CLCW Processing (DS21)
Date of Specification: September 7, 1994

The development approach to the Command Link Control Word (CLCW) Processing Building Block is to use applicable modules from the transfer frame statistics portion of the TPOCC external communications subsystem.

(Reference: TPOCC DDS for Release 10, Section 3.7.5)

1. Functional and Performance Overview

1.1 Functional Requirements Summary

The CLCW Processing Building Block shall conduct parameter stripping and conversion of the CLCW.

1.2 Performance Requirements Summary

The CLCW Processing Building Block shall be capable of processing one CLCW every second.

2. Specifications

2.1 Interface Specifications (Figure DS21-1)

The CLCW Processing Building Block shall interface with Spacecraft Communications Services, the system clock, State Manager Function, Database Building Block, Data Server Building Block, and Event Logger Building Block.

2.1.1 Input

2.1.1.1 Spacecraft Communications Services: The CLCW Processing Building Block shall receive the CLCW from Spacecraft Communications Services through a TCP/IP socket connection.

2.1.1.2 System Clock: The CLCW Processing Building Block shall receive the current time from the synchronized system clock.

2.1.1.3 State Manager Function: The CLCW Processing Building Block shall receive “terminate CLCW processing” directives from the State Manager Function.

2.1.1.3 Database Building Block: The CLCW Processing Building Block shall receive setup information from the Database Building Block.

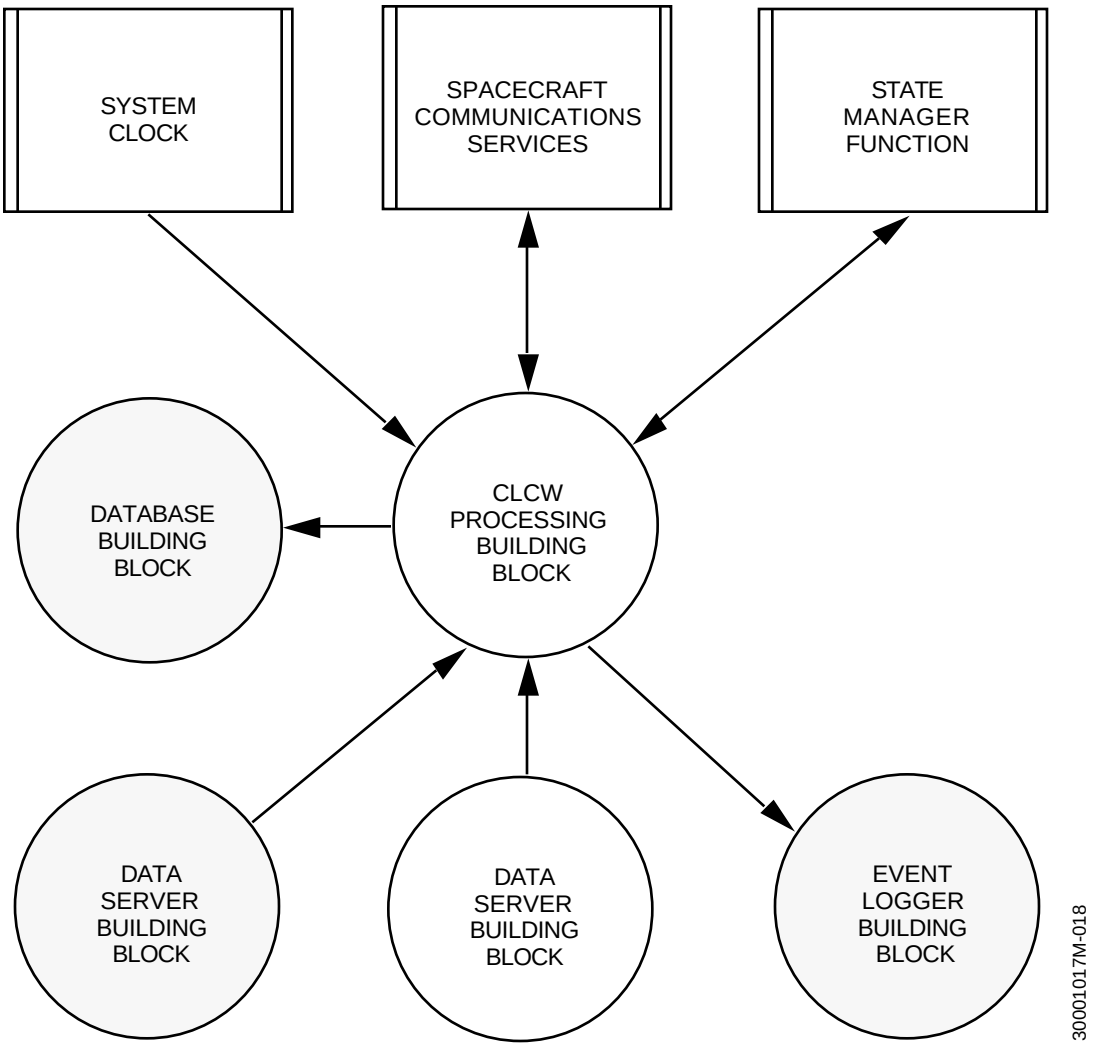

Figure DS21–1. CLCW Processing Building Block Context

2.1.2 Output

2.1.2.1 Data Server Building Block: The CLCW Processing Building Block shall provide converted CLCW parameters to the Data Server Building Block for dissemination to other building blocks.

2.1.2.2 Event Logger Building Block: The CLCW Processing Building Block shall provide error messages and other events to the Event Logger Building Block.

2.1.2.3 State Manager Function: The CLCW Processing Building Block shall provide directive status messages to the State Manager Function.

2.1.3 Standards

The CLCW Processing Building Block shall comply with the POSIX standard.

2.1.4 Interface Performance Requirements

The CLCW Processing Building Block shall receive CLCWs at a rate of one per second.

2.2 Functional Specifications

2.2.1 Functional Requirements Modifications

2.2.1.1 The CLCW Processing Building Block shall strip the bit fields from the CLCW and create separate parameters for each field.

2.2.1.2 The CLCW shall optionally convert the values in the bit fields to ASCII text or Boolean values.

2.2.2 Performance Requirements

The CLCW Processing Building Block shall process one CLCW every second.

3. Programmatic Requirements

3.1 Development Considerations

The CLCW Building Block shall be developed to comply with the ACE development environment as described in TBS.

3.2 Portability

The CLCW Processing Building Block shall compile and execute on any platform that supports POSIX and ANSI C standards.

3.3 Expandability

The CLCW Processing Building Block shall be expandable to accommodate higher data rates and to receive CLCWs from up to 255 VC IDs.

3.4 Built-In Flexibility and Customization

The location of bit fields and the conversions to be performed shall be contained in a parameter file. The ACE specific values can be found in *ACE Command and Data Handling Component Specifications* (Preliminary), January 28, 1994.

Abbreviations and Acronyms

ACE	Advanced Composition Explorer
AIS	Automated Information System
ANSI	American National Standards Institute
API	applications program interface
ASCII	American Standard Code for Information Interchange
CCSDS	Consultative Committee for Space Data Systems
CDHF	Central Data Handling Facility
CDR	critical design review
CD-ROM	compact disk read-only memory
CLCW	command link control word
COTS	commercial off-the-shelf
CRC	cyclic redundancy code
DBMS	database management system
DDF	Data Distribution Facility (also known as DDFI)
DDS	detailed design specification
DMR	detailed mission requirements
DSN	Deep Space Network
FDD	Flight Dynamic Division
FTP	File Transfer Protocol
HST	Hubble Space Telescope
ID	identifier
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
ISTP	International Solar-Terrestrial Physics
JPL	Jet Propulsion Laboratory
kbps	kilobits per second
LZP	level-zero processing
MOC	Mission Operations Center
NASA	National Aeronautics and Space Administration

Nascom	NASA Communications
NFS	Network File System
NRT	nonreal time
Pacor II	Packet Processor II Data Capture Facility
PDB	project database
PDPS	production data processing subsystem
PICS	Pacor II information and control subsystem
POSIX	Portable Open System Interconnect Executive
QAWS	quality analysis workstation subsystem
SFDU	standard formatted data unit
SQL	structured query language
TCP	Transmission Control Protocol
TDM	time-division multiplexed
TNIF	TPOCC Nascom interface
TPOCC	Transportable Payload Operations Control Center
VC	virtual channel
VCDU	virtual channel data unit
XDR	External Data Representation